

Center for Substance Abuse Treatment Opioid Treatment Data System

Digital Access to Medication Web Service Software Interface Data Definition

Version: 1.0
May 15, 2007

COTR Name:
Arlene Stanton, Ph.D., N.C.C.

Prepared for:
Substance Abuse and Mental Health Services Administration
Center for Substance Abuse Treatment
Division of Pharmacologic Therapies
1 Choke Cherry Road
Rockville, Maryland 20850

Contract Number:
270-05-0108

| | | | |
|-------------------------|--|-----------------|----------|
| Project: | CSAT Opioid Treatment Data System | | |
| Title: | D-ATM Web Service Software Interface Data Definition | Version: | 1.0 |
| Contract Number: | 270-05-0108 | Date: | 05/15/07 |

Revision History

| Date | Version | Description | Author |
|----------------|----------------|------------------------------|-----------------------|
| June 17, 2006 | 0.1 | Partial draft | Sherry Zucker |
| April 26, 2007 | 0.1 | Initial Draft Completed | Kevin H. Do |
| April 30, 2007 | 0.2 | QA/Format/Proof/Edit | Rick Camilo |
| April 30, 2007 | 0.3 | Content Edit | Sherry Zucker |
| May 08, 2007 | 0.4 | Source Code Edit | Kevin H. Do |
| May 14, 2007 | 0.5 | Final QA/Format/Proof/Edit | Rick Camilo |
| May 14, 2007 | 0.6 | Final Review | John-William DeClaris |
| May 15, 2007 | 1.0 | Initial Release as Delivered | John-William DeClaris |
| | | | |
| | | | |
| | | | |

*All Trademarks, Registered Trademarks, Service Marks, and
brand and product names used in this document
are the property of their respective owners.*

Contents

| | |
|--|-----------|
| 1. Introduction | 1 |
| 1.1 Background..... | 1 |
| 1.2 D-ATM Concept of Operations | 2 |
| 2. Software Application Programming Interface (API) | 3 |
| 2.1 Purpose of the API..... | 3 |
| 2.2 D-ATM Information to be Stored in Clinical System | 3 |
| 2.3 Transport Mechanism | 3 |
| 3. Software Development Environment..... | 3 |
| 3.1 Using Web Service Technology with the Sun Java Development Kit (JDK)..... | 4 |
| 3.1.1 Standard Java Packages Requirement..... | 4 |
| 3.1.2 Software Configuration Requirement | 4 |
| 3.1.3 Generating Web Service Client Proxy Classes | 4 |
| 3.2 Using Web Service Technology with Eclipse Java IDE | 5 |
| 3.2.1 Eclipse Software Requirement | 5 |
| 3.2.2 Generating Web Service Client Proxy Classes | 5 |
| 3.3 Using Web Service Technology with Microsoft Visual Studio 2005..... | 10 |
| 3.3.1 Generating Web Service Client Proxy Classes | 10 |
| 4. Interface Transactions | 12 |
| 4.1 Interface Transactions from Clinical System to D-ATM | 12 |
| 4.1.1 Login Authorization and Retrieving Session Ticket..... | 12 |
| 4.1.2 Medicine Order | 12 |
| 4.1.3 Completed Dosing Event..... | 12 |
| 4.2 Interface Transactions from D-ATM to Clinical System | 12 |
| 5. Data Definition for Interface Transactions | 12 |
| 5.1 Retrieving Session Ticket Transaction | 12 |
| 5.1.1 Data Transaction Fields..... | 14 |
| 5.1.2 System Message | 14 |
| 5.1.3 Java Sample Code | 15 |
| 5.1.4 VB.NET Sample Code | 15 |
| 5.1.5 C# Sample Code | 16 |
| 5.2 Medicine Order Transactions..... | 17 |
| 5.2.1 Create Medicine Order | 17 |
| 5.2.2 Update Medicine Order | 26 |
| 5.2.3 Delete Medicine Order..... | 32 |
| 5.3 Dosing Event Transactions | 38 |
| 5.3.1 Create Completed Dosing Event | 38 |
| 5.3.2 Delete Completed Dosing Event | 47 |

| | | |
|-------|---|----|
| 5.4 | Interface Transactions from D-ATM to Clinical System | 54 |
| 5.4.1 | Dose Administered to Patient with a Current Medicine Order at the Requesting OTP by Another OTP | 54 |
| 5.4.2 | Guest Retrieval Events and Doses Administered to a Patient with a Current Medicine Order by Another OTP | 66 |

List of Figures

| | | |
|-----------|---|----|
| Figure 1. | Eclipse SDK – New Java Class..... | 6 |
| Figure 2. | Eclipse SDK – New Web Service Client..... | 7 |
| Figure 3. | Eclipse SDK – Web Service Definition | 8 |
| Figure 4. | Eclipse SDK – D-ATM Web Service Package..... | 9 |
| Figure 5. | Microsoft Visual Studio – Web Service Reference..... | 10 |
| Figure 6. | Microsoft Visual Studio – Web Service URL | 11 |
| Figure 7. | Microsoft Visual Studio – D-ATM Web Service Package | 11 |

List of Tables

| | | |
|-----------|--|----|
| Table 1. | Interface Transactions – Data Transaction Field Names | 14 |
| Table 2. | Interface Transactions – System Message | 14 |
| Table 3. | Create Medicine Order – Data Transaction Fields | 18 |
| Table 4. | Medicine Order Transactions – System Messages | 21 |
| Table 5. | Update Medicine Order – Data Transaction Fields | 27 |
| Table 6. | Update Medicine Order – System Messages | 28 |
| Table 7. | Delete Medicine Order – Data Transaction Fields..... | 33 |
| Table 8. | Delete Medicine Order – System Messages | 34 |
| Table 9. | Create Completed Dosing Event – Data Transaction Fields | 39 |
| Table 10. | Create Completed Dosing Event – System Messages..... | 41 |
| Table 11. | Delete Completed Dosing Event – Data Transaction Fields | 48 |
| Table 12. | Delete Completed Dosing Event – System Messages | 50 |
| Table 13. | Dose Administered to Patient w/ Current Medicine Order at Requesting OTP by Another OTP – Data Transaction Fields | 55 |
| Table 14. | Dose Administered to Patient w/ Current Medicine Order at the Requesting OTP by Another OTP – System Messages..... | 57 |
| Table 15. | Guest Retrieval Events and Doses Administered to a Patient w/ Current Medicine Order by Another OTP – Data Transaction Fields..... | 68 |
| Table 16. | Guest Retrieval Events and Doses Administered to a Patient w/ Current Medicine Order by Another OTP – System Messages | 70 |

Digital Access to Medication Web Service

Software Interface Data Definition

1. Introduction

1.1 Background

The terrorist attacks on September 11, 2001 had an immediate and profound impact on New York City's opioid treatment system. One opioid treatment program (OTP), in the immediate vicinity of the World Trade Center, was destroyed. Several other OTPs were closed for days or even weeks. In all, it is estimated some 1,000 patients were displaced. Despite the chaos and terror surrounding this tragic event, patients sought treatment and staff and administrators in the New York metropolitan area kept other OTPs open, serving their own as well as these displaced patients. In August, 2005, history seemed to repeat itself as Katrina, the costliest and one of the deadliest hurricanes in United States history, wreaked catastrophic damage in Louisiana, Mississippi, and Alabama. In the days following, 80% of the city of New Orleans was flooded as the levees were breached. All seven New Orleans OTPs were shut down for an extended period of time, and months later, many remain closed. As with 9-11, staff at other programs were forced to make critical dosing decisions, in many cases based upon the information provided by the patients themselves.

Immediately after the events of 9–11, working with a number of key stakeholders from New York, Connecticut, and New Jersey, the Center for Substance Abuse Treatment (CSAT), a component of the Substance Abuse and Mental Health Services Administration (SAMHSA), developed a feasibility and planning study focused on developing a system that would ensure patients could safely and appropriately be dosed in the event that a disaster or other event resulted in closure of one or more treatment sites. In Fall, 2005, SAMHSA/CSAT funded a project to implement a limited pilot of the system, starting in the metropolitan New York area. From the beginning, the project has been guided by four principles: simplicity, affordability, acceptability and confidentiality.

Called D-ATM (Digital Access to Medication), the project entails development and implementation of a pilot system to support the retrieval of limited patient dosage information in the event of both major disasters and also more common disruptions in service due to snow storms, power failures and other localized events. The pilot system uses a biometric device, such as a finger scanner, to store recent dosage information to a centralized database. Should a patient need to seek treatment at a program other than the one where he/she is normally enrolled, the 'guest' OTP will be able to verify that the person is a patient, and then retrieve a report on that patient's medicine order and dosage simply by scanning the patient's finger.

The current project involves development of the necessary infrastructure, followed by a pilot test of D-ATM with 50 or more OTPs to determine the system's effectiveness in supporting OTPs disaster preparedness and recovery activities for any discontinuity in service. Following the pilot, the system will be reviewed to determine its effectiveness and ability to support a national implementation.

1.2 D-ATM Concept of Operations

The purpose of D-ATM is to serve as a disaster response system by providing data for guest OTPs to assist patients who need treatment during service discontinuity of their home OTPs. Such discontinuity could include minor disruptions, not just large-scale disasters such as Hurricane Katrina or the events of September 11, 2001. From the patients' perspective, their personal disaster is the inability to receive medications regardless of the cause of service discontinuity.

A patient who cannot get to his/her 'home' OTP can go to a 'guest' OPT participating in the system. The guest OPT would identify the patient to the system by scanning the patient's finger. D-ATM would then display patient's medicine order and dosing history over the last 90 days. If internet access is unavailable, the guest OTP could still scan the patient's finger to retrieve the patient's D-ATM ID since this information will be stored locally. Then the OTP staff member would dial a toll-free phone number and present his/her identification and the patient's D-ATM ID to a support staff person, who would access D-ATM on behalf of the OTP. The patient's medicine order and dosage information would then be faxed directly from D-ATM to the guest OTP.

For D-ATM to provide the guest OPT with the information necessary for determining the proper dose, the following information must be captured from the patient's home (and guest) OTP:

- Patient's home OTP name and contact;
- Patient's current medicine order;
- Daily dosing events for the past 90 days including dispensing date, dosage amount, and medicine.

The major functional components of D-ATM for the home or guest OTP where the patient is enrolled are:

1. Enrolling the patient – capturing the patient's finger scan and associating a D-ATM patient ID with the data points ('minutiae') of the finger scan;
2. Entering the patient's medicine order by capturing the expiration date, dose amount, if the dose is varied and by how much over what period of time, days for take-home vs. clinic doses, date of patient's next appointment, and other pertinent data;
3. Recording patient's daily dosing events by capturing the dispensing date, dosage amount, whether clinic visit or take home, and any comments;
4. Retrieving patient's medicine order and dosing history from D-ATM;

D-ATM facilitates communication between the OPT treating the same patient. When a guest OTP performs a search for dosing information for a patient, D-ATM alerts the home OTP (except after the patient is enrolled at the guest OTP). More importantly, when a guest OTP provides a dose to a guest patient, the other OTPs with a current medicine order for this patient will be notified.

D-ATM is being developed to capture the required data two ways:

1. As a stand-alone system;
2. As a back-end system with a software interface (API) with the OTP clinical systems.

With the stand-alone system, the OTP must enter the patient's enrollment, medicine order, and record the patient's dosing events using the D-ATM user interface and application. Since most of this information is already being captured by OTPs in their local clinical systems, it puts a burden on OTP staff to perform duplicate data entry. For this reason, a software interface is planned to facilitate the communication of data between D-ATM and the OTP clinical systems.

2. Software Application Programming Interface (API)

2.1 Purpose of the API

Many, if not most, of the OTPs use COTS (commercial off-the-shelf software) to manage the clinic and to track patient dosing information. It is not the intention of D-ATM to compete with these robust software packages and proprietary systems. Nor is it practical to require OTP staff to perform duplicate data entry in their clinical system and in D-ATM. Rather, the strategy is for COTS vendors to build into their software the ability to communicate data to and receive data from D-ATM. This approach eliminates the need for D-ATM to reproduce functionality already included in clinical systems software.

The purpose of this document is to define the transactions and data that must be communicated between D-ATM and the clinical systems. For D-ATM to be successful there must be coordination and cooperation with the major vendors of OTP clinical systems.

2.2 D-ATM Information to be Stored in Clinical System

There is some information that the clinical system must store to communicate with D-ATM:

1. The D-ATM patient ID, which is based on the patient's biometric ID (data points from finger scan);
2. Data elements required by D-ATM (see transactions);
3. An indication of which records in the clinical system were communicated to D-ATM, the date of transfer, and possibly the status of the transfer (e.g., successful update, rejection, error, etc.).

2.3 Transport Mechanism

This document is being distributed as a draft to solicit comments and suggestions from the major vendors of OTP clinical software. The method of data transport will be through using the D-ATM API Web Service.

3. Software Development Environment

A proper development environment will be needed to interact with the D-ATM API Web Service. This section provides the requirements and instructions for integrating Web Service technology to both Sun Java Development Kit (JDK) applications, as well as software built using the Eclipse platform (an open source Java IDE) and Microsoft Visual Studio 2005.

3.1 Using Web Service Technology with the Sun Java Development Kit (JDK)

3.1.1 Standard Java Packages Requirement

The D-ATM API Web Service has been successfully tested on Java clients using the following required software packages:

1. Sun Java SE JDK 6 Update 1
2. Sun JavaMail API 1.4
3. Sun JavaBeans Activation Framework 1.1
4. Apache SOAP 2.3.1
5. Apache Xerces 1.4.4
6. Apache Axis 1.4

3.1.2 Software Configuration Requirement

- Ensure the CLASSPATH environment variable includes the proper jar files needed for Web Service development. Assuming the usage of the java packages from above, the CLASSPATH should include:

```
<INSTALL FOLDER>\soap-2_3_1\lib\soap.jar; <INSTALL FOLDER>\xerces-1_4_4\xerces.jar;  
<INSTALL FOLDER>\javamail-1.4\mail.jar; <INSTALL FOLDER>\jaf-1.1\activation.jar; <INSTALL  
FOLDER>\soap-2_3_1; <INSTALL FOLDER>\axis-1_4\lib\axis.jar; <INSTALL FOLDER>\axis-  
1_4\lib\jaxrpc.jar; <INSTALL FOLDER>\axis-1_4\lib\saaj.jar; <INSTALL FOLDER>\axis-  
1_4\lib\commons-logging-1.0.4.jar; <INSTALL FOLDER>\axis-1_4\lib\commons-discovery-0.2.jar;  
<INSTALL FOLDER>\axis-1_4\lib\wsdl4j-1.5.1.jar;;
```

- Ensure the JAVA_HOME environment variable is set to the JDK or Java SDK installation folder (i.e. <INSTALL FOLDER>\Java\jre1.6.0_01).
- Ensure the Path environment variable includes binary folder for the JDK or Java SDK such as <INSTALL FOLDER>\Java\jdk1.6.0_01\bin.

The <INSTALL FOLDER> tag represents the installation directory path (i.e. “C:\Program Files”).

3.1.3 Generating Web Service Client Proxy Classes

Java proxy classes will be used to consume the D-ATM API Web Service. Apache AXIS provides a tool (WSDL2Java) to convert D-ATM API WSDL specification file to Java source code. The command to execute the tool with the D-ATM API Web Service is:

```
java org.apache.axis.wsdl.WSDL2Java https://pilot.d-atm.org/api_ws/datmAPIWS.asmx?WSDL
```

The tool will generate the following Java classes within the “.\org\d_atm\pilot\api_ws” folder:

- DATMAPIWS.java
- DATMAPIWSLocator.java
- DATMAPIWSSoap.java
- DATMAPIWSSoapStub.java

- DATMAPIWSSoap12Stub.java
- DATMInfoWSC.java

The Java client accessing the D-ATM API Web Service will require these proxy classes.

3.2 Using Web Service Technology with Eclipse Java IDE

3.2.1 Eclipse Software Requirement

The following standard Java packages and Eclipse components were included during the creation of proxy classes for the D-ATM API Web Service:

1. Sun Java SE JDK 6 Update 1
2. Eclipse SDK 3.2
3. Web Tools Platform (WTP) 1.5.3
4. Eclipse Modeling Framework (EMF) 2.2.2
5. Graphic Editing Framework (GEF) 3.2.2
6. Visual Editor (VE) 1.2
7. Apache Tomcat 5.5

3.2.2 Generating Web Service Client Proxy Classes

1. Right-click **Project**. (See Figure 1.)
2. Under the **New** command, select **Other**.

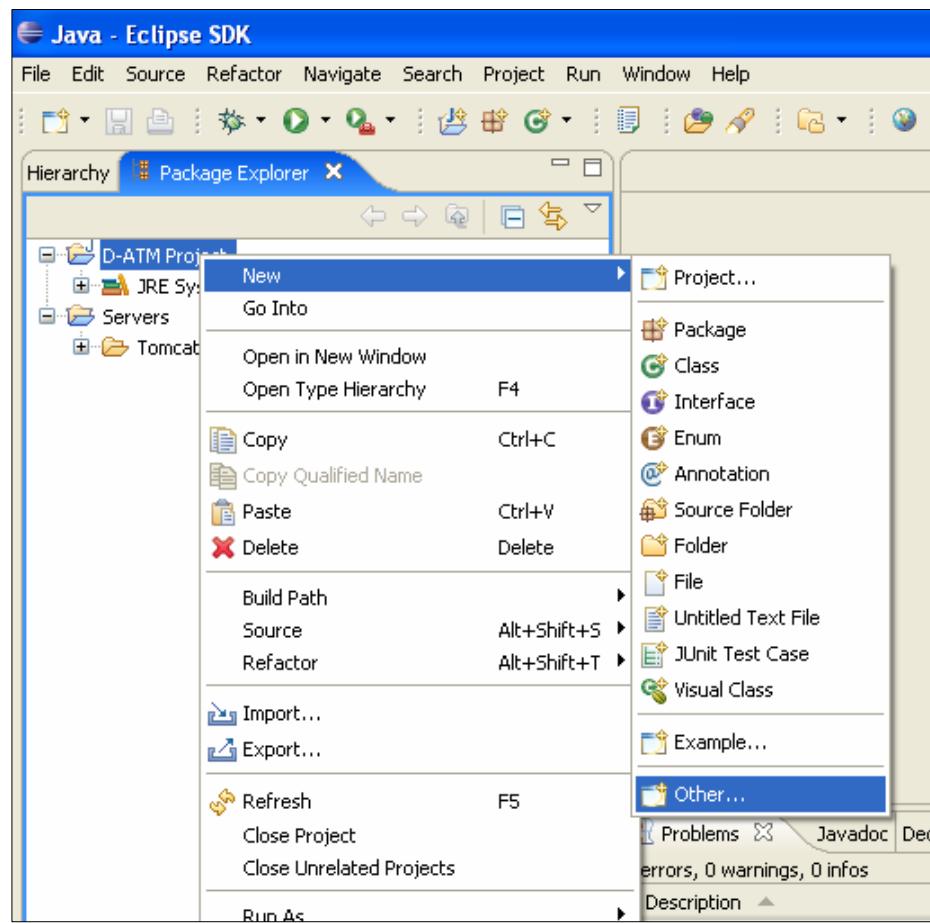


Figure 1. Eclipse SDK – New Java Class

3. Under **Web Services**, select **Web Service Client**, and then click <Next>. (See Figure 2.)

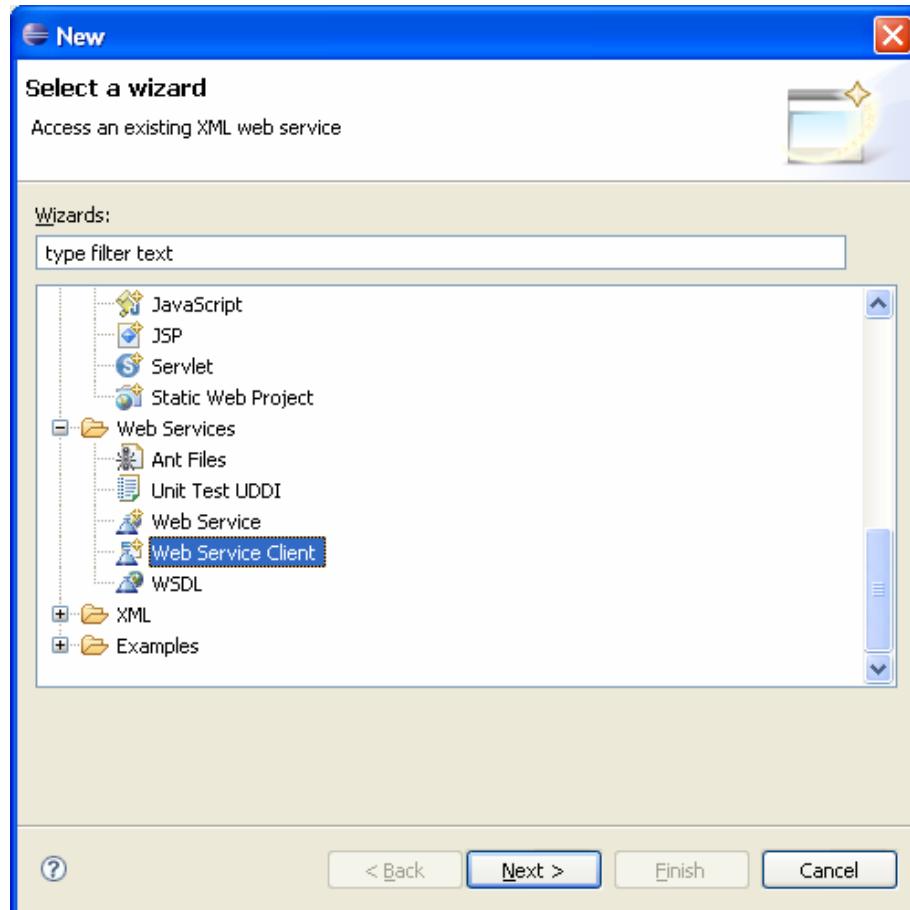


Figure 2. Eclipse SDK – New Web Service Client

4. Enter [Service Definition] https://pilot.d-atm.org/api_ws/datmAPIWS.asmx?WSDL, and then click <Finish>. (See Figure 3.)

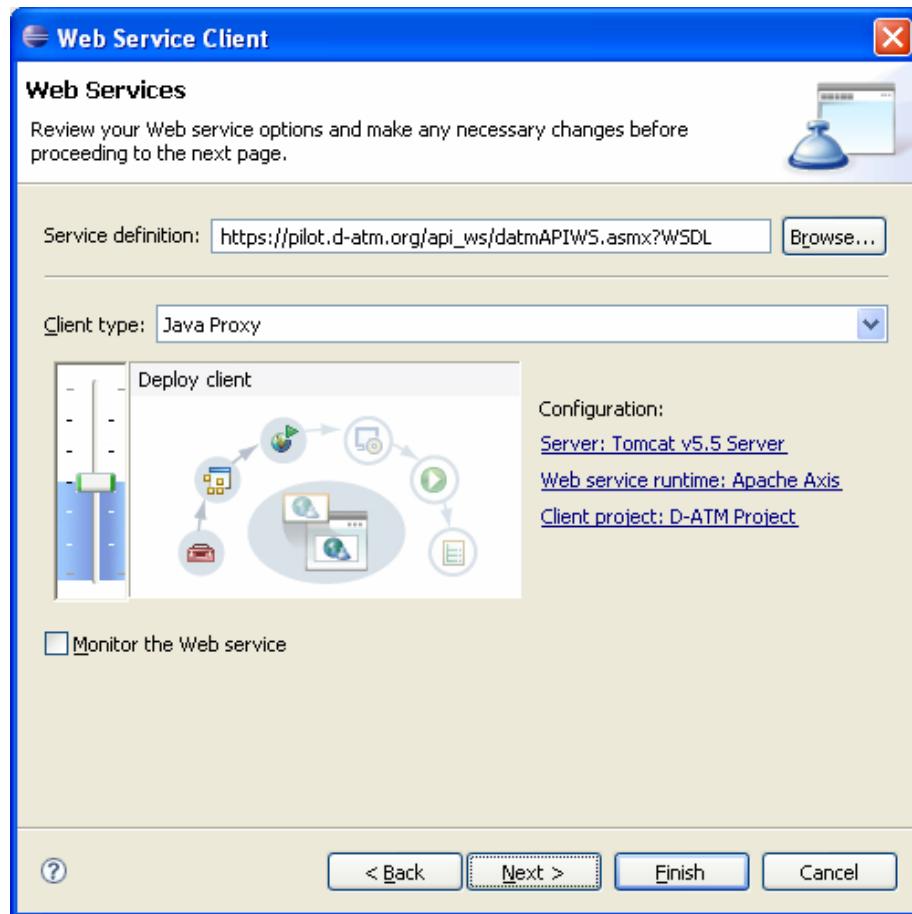


Figure 3. Eclipse SDK – Web Service Definition

5. The Java proxy classes are generated and stored inside the **org.d_atm.pilot.api_ws** package. (See Figure 4.)

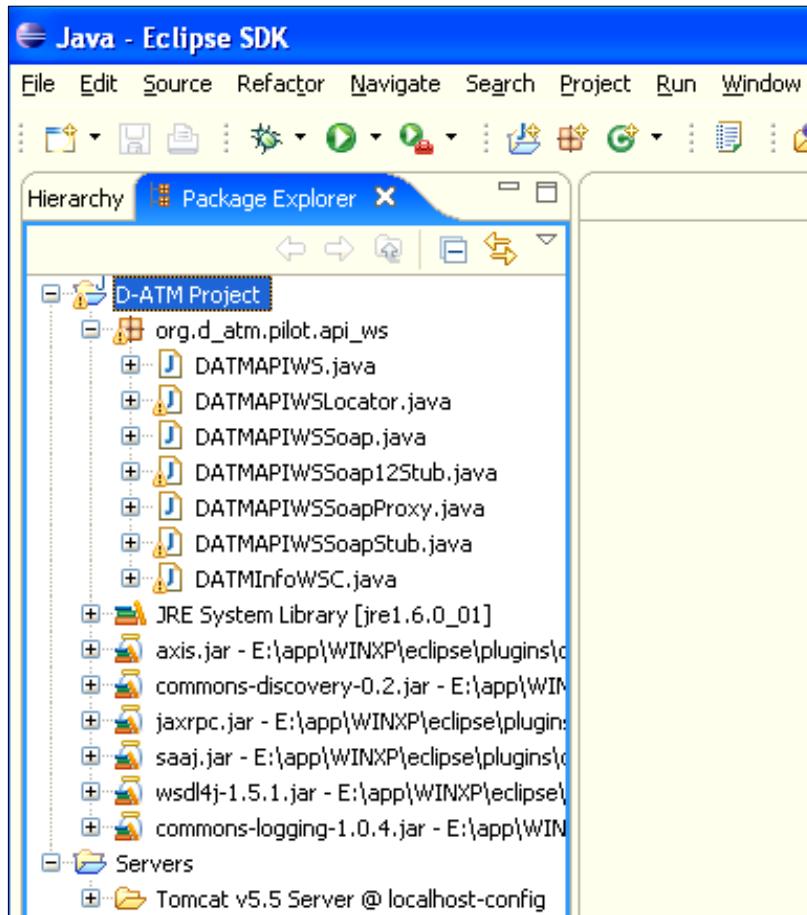


Figure 4. Eclipse SDK – D-ATM Web Service Package

3.3 Using Web Service Technology with Microsoft Visual Studio 2005

3.3.1 Generating Web Service Client Proxy Classes

1. On the **main** menu, click the **Project** command, and then select **Add Web Reference**. (See Figure 5.)

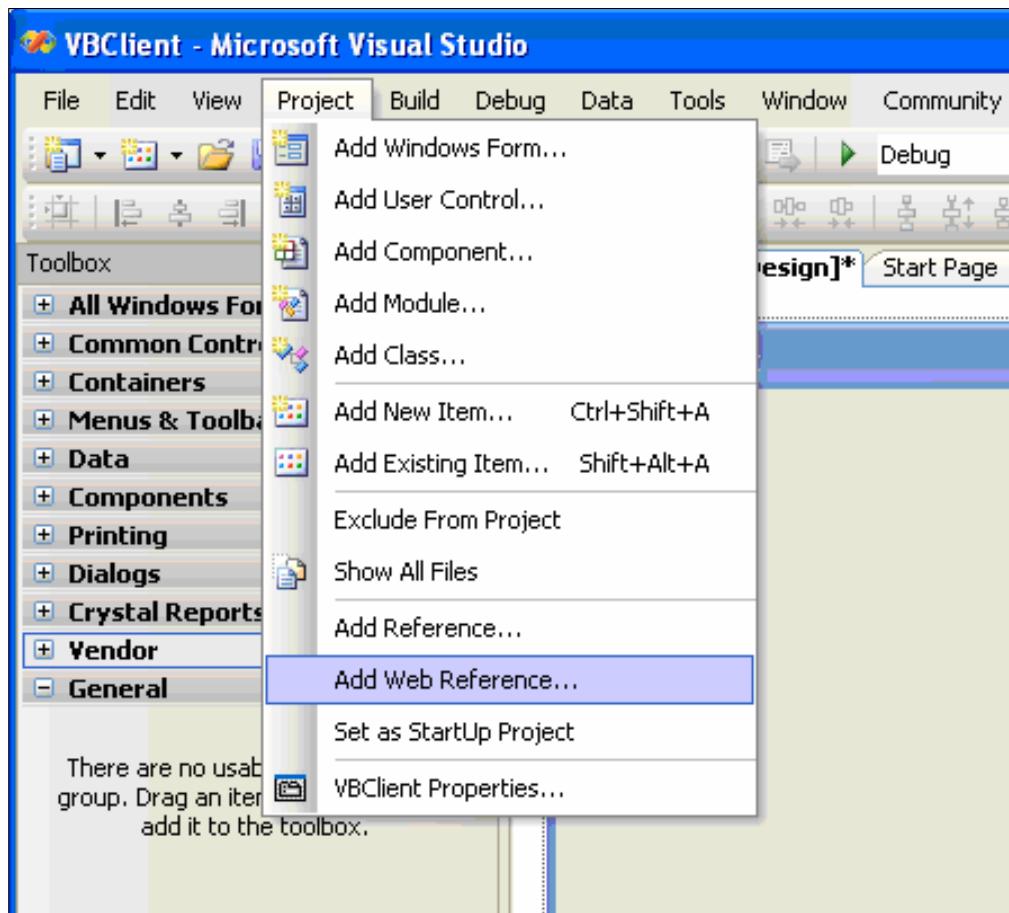


Figure 5. Microsoft Visual Studio – Web Service Reference

2. Enter the [URL] https://pilot.d-atm.org/api_ws/datmAPIWS.asmx. (See Figure 6.)

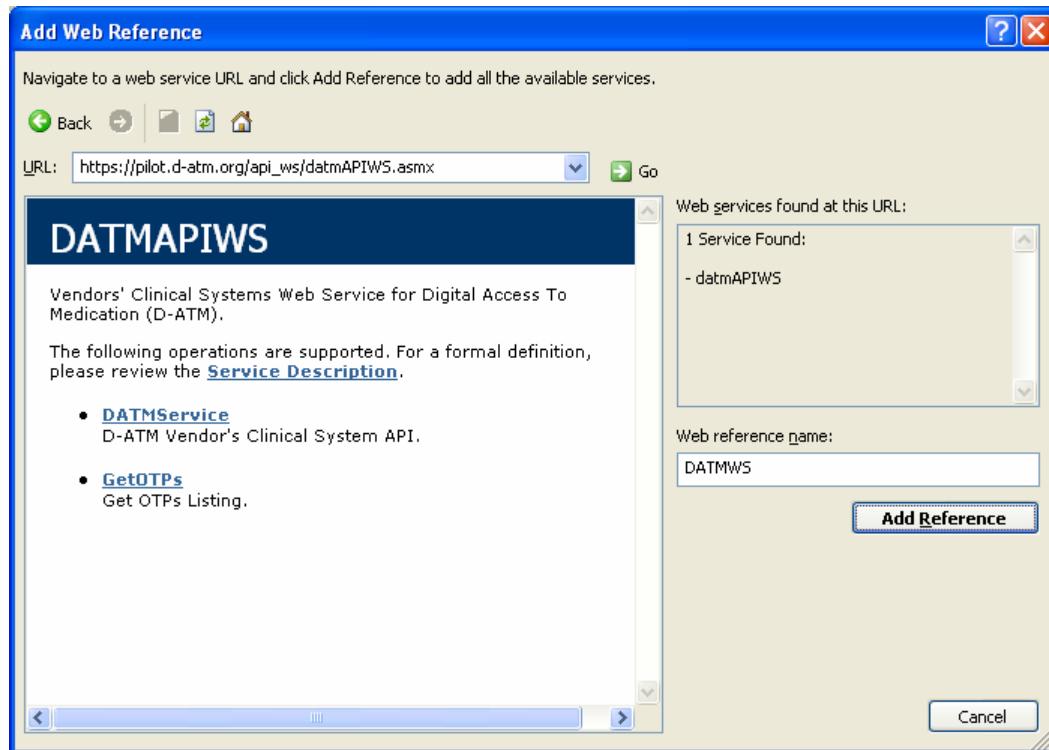


Figure 6. Microsoft Visual Studio – Web Service URL

3. Type DATMWS, and then click <Add Reference>.
4. The proxy classes are generated and stored under the **DATMWS** package as shown in Figure 7.

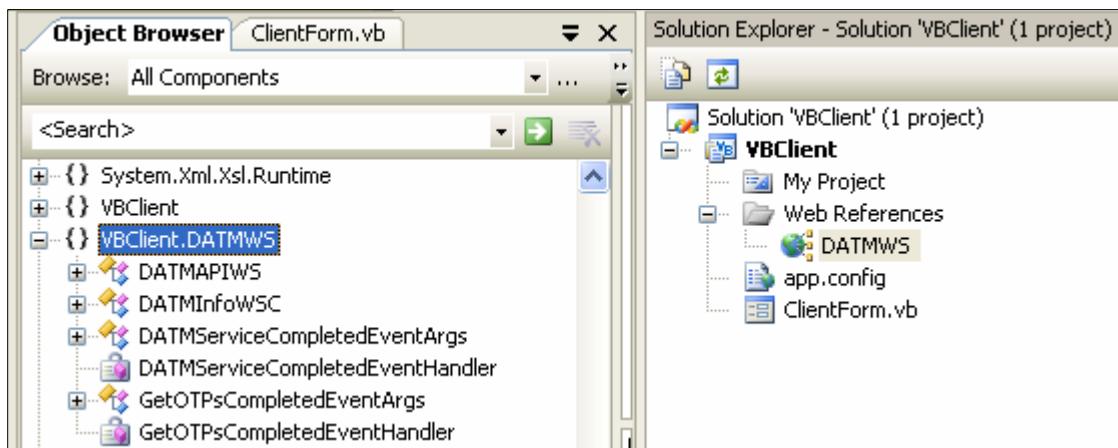


Figure 7. Microsoft Visual Studio – D-ATM Web Service Package

4. Interface Transactions

The following transactions have been defined.

4.1 Interface Transactions from Clinical System to D-ATM

4.1.1 Login Authorization and Retrieving Session Ticket

- Access web service with proper D-ATM login and password;
- Retrieving GUID session ticket for future web service calls;

4.1.2 Medicine Order

- Create new medicine order for a patient at an OTP;
- Update existing medicine order;
- Remove medicine order.

4.1.3 Completed Dosing Event

- Create completed dosing event for patient at an OTP;
- Remove completed dosing event.

4.2 Interface Transactions from D-ATM to Clinical System

- Dose administered to a patient with a current medicine order at requesting OTP by another OTP;
- Retrieval by another OTP of dosing information for a home patient at the

5. Data Definition for Interface Transactions

5.1 Retrieving Session Ticket Transaction

In addition to the standard Login and Password authentication, D-ATM also implements a “ticket system,” which provides Web service authentication for succeeding web service calls at the application level.

The steps associated with the ticket system technique are:

1. Passing the Login ID and Password through the ticket retrieval Web Service method initially authenticates the user.
2. The Web service generates a GUID ticket (or session identifier) that relates to the user account.
3. The GUID ticket is stored in a database table and as a cache object on the Web server.
4. The GUID ticket is issued to the client.
5. The GUID ticket will be stored in a global variable or session object within the client application.

6. The GUID ticket is used on subsequent Web service calls, eliminating the need for the user to repeatedly send his/her full credentials (Login ID and Password) each time the client makes a request to the Web service.
7. The ticket is valid for a set period of time, after a period of inactivity the session expires and the user must log into the system again to request a new ticket.

5.1.1 Data Transaction Fields

Table 1. Interface Transactions – Data Transaction Field Names

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|---------------------------|---|------------------|-----------|----------|---|
| INPUT TRANSACTION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| TTYP | Transaction type for medicine order | "TK" | String | Yes | |
| CMD | Command to create a medicine order. | "R" | String | Yes | |
| OUTPUT TRANSACTION | | | | | |
| GULd | Session Ticket | | String | | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| Timeout | Session Timeout in minutes | | String | | |

5.1.2 System Message

Table 2. Interface Transactions – System Message

| Message ID | Message Name | Message Text |
|------------|---------------|--|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |

5.1.3 Java Sample Code

```
import org.d_atm.pilot.api_ws.*;  
  
public class GetSession  
{  
    public static void main(String [] args)  
    {  
        try  
        {  
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();  
            inDATMInfo.setLogin("OTP1");  
            inDATMInfo.setPassword("Test1234");  
            inDATMInfo.setTTYP("TK");  
            inDATMInfo.setCMD("R");  
  
            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();  
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();  
  
            DATMInfoWSC outDATMInfo = new DATMInfoWSC();  
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);  
  
            String message = "";  
  
            if (outDATMInfo.getMessageId().length() > 0) {  
                message = outDATMInfo.getMessageText();  
            }  
            else {  
                message = message + "Session GUID: " + outDATMInfo.getGUID() + "\n";  
                message = message + "Session Timeout: " + outDATMInfo.getTimeout();  
            }  
  
            System.out.println(message);  
        }  
        catch(Exception e)  
        {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

5.1.4 VB.NET Sample Code

```
Imports VBClient.DATMWS  
Imports VBClient.ClientForm  
Imports System  
Imports System.IO  
Imports System.Data  
Imports System.Xml
```

```
Public Class ClientForm
```

```
Private Sub btnGetSession_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles btnGetSession.Click  
    Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()
```

```

Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()

inDATMInfo.Login = "OTP1"
inDATMInfo.Password = "Test1234"

inDATMInfo.TTYP = "TK"
inDATMInfo.CMD = "R"

Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()

outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

If Not (outDATMInfo.MessageId = "") Then
    txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, System.Environment.NewLine)
Else
    If Not (outDATMInfo.GUID = "") Then
        txtMessage.Text = "Session GUID: " & _
            outDATMInfo.GUID.Replace(ControlChars.Lf, System.Environment.NewLine) & _
            System.Environment.NewLine
        txtMessage.Text = txtMessage.Text & "Session Timeout: " & _
            outDATMInfo.Timeout.Replace(ControlChars.Lf, System.Environment.NewLine) & _
            System.Environment.NewLine
    End If
End If

wsDATMWebService = Nothing
End Sub

```

End Class

5.1.5 C# Sample Code

```

using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void btnGetSession_Click(object sender, EventArgs e)
        {
            DATMAPIWS wsDATMWebService = new DATMAPIWS();

```

```
DATMInfoWSC inDATMInfo = new DATMInfoWSC();
inDATMInfo.Login = "OTP1";
inDATMInfo.Password = "Test1234";
inDATMInfo.TTYP = "TK";
inDATMInfo.CMD = "R";

DATMInfoWSC outDATMInfo = new DATMInfoWSC();
outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);

if (outDATMInfo.MessageId.Length > 0)
{
    txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
}
else
{
    if (outDATMInfo.GUID.Length > 0)
    {
        txtMessage.Text = "Session GUID: " +
            outDATMInfo.GUID.Replace("\n", System.Environment.NewLine) +
            System.Environment.NewLine;
        txtMessage.Text = txtMessage.Text + "Session Timeout: " +
            outDATMInfo.Timeout.Replace("\n", System.Environment.NewLine) +
            System.Environment.NewLine;
    }
}
wsDATMWebService = null;
}
```

5.2 Medicine Order Transactions

Using the API, medicine orders can be created, updated, or removed.

5.2.1 Create Medicine Order

A patient must be enrolled at the OTP, either as a home or guest patient, before a medicine order can be created at that OTP for that patient.

There can be only one current medicine order for a patient enrollment at an OTP; however, there may be multiple current medicine orders for the patient at different OTPs. If there's a request to create a new medicine order and that patient already has a current medicine order, the current medicine order will be expired and the new medicine order created.

5.2.1.1 Data Transaction Fields

Table 3. Create Medicine Order – Data Transaction Fields

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|----------------------------|--|-----------------------------------|-----------|----------|---|
| INPUT AUTHORIZATION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| OR | | | | | |
| GUid | Session Ticket | | String | Yes | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| INPUT TRANSACTION | | | | | |
| TTYP | Transaction type for medicine order | "MO" | String | Yes | |
| CMD | Command to create a medicine order. | "C" | String | Yes | |
| CLMOID | Clinical system medicine order ID | | String | Yes | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from the finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | D-ATM patient ID ten-digit number | String | Yes | Example: "7124510603" |
| OTPId | The SAMHSA OTP ID creating, updating, or deleting the medicine order | Valid 8-digit SAMHSA OTP ID | String | Yes | Example: LESC's OTP ID is "NY10153M" |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|---------------|--|--|-----------|----------|---|
| EffectDt | Medicine order effective date. The date that the medicine order goes into effect. Note that in D-ATM it is possible to create a medicine order in advance of when it goes into effect. | | String | Yes | Format: mm/dd/yyyy Example: "04/05/2007" |
| ExpiraDt | Medicine order expiration date. The date after which the medicine order is no longer valid. | | String | Yes | Format: mm/dd/yyyy Example: "04/05/2007" |
| NxtapptDt | Next medical appointment date for continuing medicine order. The date of the next appointment related to the patient's medicine order for a controlled substance (e.g., Methadone), and not for other medications. | | String | | Format: mm/dd/yyyy Example: "04/05/2007" |
| Medicine | Code for Methadone, Buprenorphine, or other controlled substance | "1" = Methadone "2" = Subutex "3" = Suboxone "4" = Naloxone | String | Yes | |
| DoseType | Code for single or split dose. | "1" = Single dose "2" = Split dose | String | Yes | |
| DoseAmt | If the dose is varied, this is the starting dosage amount. If dose isn't varied, then this is the daily dose amount. The amount in milligrams of a daily dose of the controlled substance. | | String | Yes | Example: "25", "20.50", etc... |
| VaryDoseAmt | If the dose is to be varied, this is the positive or negative number of milligrams to increase or decrease the starting dose amount in a patient's medicine order. If no sign, then a positive number is assumed. | | String | | Example: "5", "-5", etc... |
| VaryPeriodNum | If the dose is to be varied, then this is the number of time period units (e.g., weeks, doses) over which the dose amount is to be varied by the specified amount | | String | | |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|--------------------|---|---|-----------|----------|--------------|
| VaryPeriodUnit | If the dose is to be varied, then this is the time period unit (e.g., day, week, dose) over which the dose amount is to be varied by the specified amount | “D” = Daily “W” = Weekly “M” = Monthly “S” = Single Dose | String | | |
| Reason | The reason for the Patient’s medicine order dose amount. | | String | | |
| TakehomeNum | The approved number of take home doses that a patient may receive in a given time period (e.g., week, medicine order). | | String | | |
| TakehomePeriodUnit | The unit of time for which the number of take home doses (TakehomeNum) is approved (e.g., ‘per week’ or ‘per medicine order’). | “W” = Per Week “O” = Per Medicine Order | String | | |
| Comment | Medicine Order Notes/Comment. | | String | | |
| MonVisitType | If Monday’s dose is given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | | |
| TueVisitType | If Tuesday’s dose is given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | | |
| WedVisitType | If Wednesday’s dose is given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | | |
| ThuVisitType | If Thursday’s dose is given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | | |
| FriVisitType | If Friday’s dose is given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | | |
| SatVisitType | If Saturday’s dose is given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | | |
| SunVisitType | If Sunday’s dose is given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | | |

5.2.1.2 System Messages

Table 4. Medicine Order Transactions – System Messages

| Message ID | Message Name | Message Text |
|------------|--|--|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |
| 1002 | D-ATM Patient Not Enrolled At OTP To Create/Update/Delete Medicine Order | A medicine order transaction was attempted for a patient that wasn't enrolled in the OTP; therefore, the request was ignored. Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |
| 1003 | Create D-ATM Medicine Order That Already Exists | An attempt was made to create a new medicine order for a patient that already had a current medicine order at the OTP. Therefore, the expiration date on the existing medicine order was set to today's date and a new medicine order was created. Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |
| 1004 | Medicine Order Transaction Failed | Request to create/update/delete medicine order in D-ATM failed. Error(s): [List of Errors] Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |

| Message ID | Message Name | Message Text |
|------------|--------------------------------------|--|
| 1010 | Medicine Order Transaction Completed | A medicine order in D-ATM was created/updated/deleted. Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |

5.2.1.3 Java Sample Code

```
import org.d_atm.pilot.api_ws.*;  
  
public class CreateMO  
{  
    public static void main(String [] args)  
    {  
        try  
        {  
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();  
  
            inDATMInfo.setLogin("OTP1");  
            inDATMInfo.setPassword("Test1234");  
  
            // Or use a given session ticket GUID  
            // inDATMInfo.setGUID("A5ED32EF-D6D3-46E2-B3AD-6B8022F20875");  
  
            inDATMInfo.setTTYP("MO");  
            inDATMInfo.setCMD("C");  
            inDATMInfo.setCLMOID("257703-15");  
            inDATMInfo.setDATMPatientId("7124510603");  
            inDATMInfo.setOTPID("NY60001M");  
            inDATMInfo.setEffectDt("04/10/2007");  
            inDATMInfo.setExpiraDt("05/14/2007");  
            inDATMInfo.setNxtapptDt("05/15/2007");  
            inDATMInfo.setMedicine("1");  
            inDATMInfo.setDoseType("1");  
            inDATMInfo.setDoseAmt("25");  
            inDATMInfo.setVaryDoseAmt("5");  
            inDATMInfo.setVaryPeriodNum("3");  
            inDATMInfo.setVaryPeriodUnit("D");  
            inDATMInfo.setReason("Initial Treatment.");  
            inDATMInfo.setTakehomeNum("4");  
            inDATMInfo.setTakehomePeriodUnit("W");  
            inDATMInfo.setComment("Ready for dosing.");  
            inDATMInfo.setMonVisitType("C");  
            inDATMInfo.setTueVisitType("C");  
            inDATMInfo.setWedVisitType("C");  
            inDATMInfo.setThuVisitType("C");  
            inDATMInfo.setFriVisitType("C");  
            inDATMInfo.setSatVisitType("H");  
            inDATMInfo.setSunVisitType("H");  
  
            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();  
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();  
  
            DATMInfoWSC outDATMInfo = new DATMInfoWSC();  
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);  
  
            String message = "";  
  
            if (outDATMInfo.getMessageId().length() > 0) {  
                message = outDATMInfo.getMessageText();  
                System.out.println(message);  
            }  
        }  
    }  
}
```

```
        }
    }
} catch(Exception e)
{
    System.out.println(e.getMessage());
}
}
```

5.2.1.4 VB.NET Sample Code

```
Imports VBClient.DATMWS  
Imports VBClient.ClientForm  
Imports System  
Imports System.IO  
Imports System.Data  
Imports System.Xml
```

Public Class ClientForm

```
Private Sub btnCreateMO_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnCreateMO.Click
    Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()
    Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()
    inDATMInfo.Login = "OTP1"
    inDATMInfo.Password = "Test1234"
    ' Or use a given session ticket GUID
    ' inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875"
    inDATMInfo.TTYP = "MO"
    inDATMInfo.CMD = "C"
    inDATMInfo.CLMOID = "257703-15"
    inDATMInfo.DATMPatientId = "7124510603"
    inDATMInfo.OPTId = "NY60001M"
    inDATMInfo.EffectDt = "04/10/2007"
    inDATMInfo.ExpiraDt = "05/14/2007"
    inDATMInfo.NxtapptDt = "05/15/2007"
    inDATMInfo.Medicine = "1"
    inDATMInfo.DoseType = "1"
    inDATMInfo.DoseAmt = "25"
    inDATMInfo.VaryDoseAmt = "5"
    inDATMInfo.VaryPeriodNum = "3"
    inDATMInfo.VaryPeriodUnit = "D"
    inDATMInfo.Reason = "Initial Treatment."
    inDATMInfo.TakehomeNum = "4"
    inDATMInfo.TakehomePeriodUnit = "W"
    inDATMInfo.Comment = "Ready for dosing."
    inDATMInfo.MonVisitType = "C"
    inDATMInfo.TueVisitType = "C"
    inDATMInfo.WedVisitType = "C"
    inDATMInfo.ThuVisitType = "C"
    inDATMInfo.FriVisitType = "C"
    inDATMInfo.SatVisitType = "H"
```

```

inDATMInfo.SunVisitType = "H"

Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()

outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

If Not (outDATMInfo.MessageId = "") Then
    txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, System.Environment.NewLine)
End If

wsDATMWebService = Nothing
End Sub

End Class

```

5.2.1.5 C# Sample Code

```

using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void btnCreateMO_Click(object sender, EventArgs e)
        {
            DATMAPIWS wsDATMWebService = new DATMAPIWS();

            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.Login = "OTP1";
            inDATMInfo.Password = "Test1234";

            // Or use a given session ticket GUID
            // inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875";

            inDATMInfo.TTYP = "MO";
            inDATMInfo.CMD = "C";
            inDATMInfo.CLMOID = "257703-15";
            inDATMInfo.DATMPatientId = "7124510603";
            inDATMInfo.OTPlId = "NY60001M";
            inDATMInfo.EffectDt = "04/10/2007";
            inDATMInfo.ExpiraDt = "05/14/2007";
        }
    }
}

```

```
inDATMInfo.NxtapptDt = "05/15/2007";
inDATMInfo.Medicine = "1";
inDATMInfo.DoseType = "1";
inDATMInfo.DoseAmt = "25";
inDATMInfo.VaryDoseAmt = "5";
inDATMInfo.VaryPeriodNum = "3";
inDATMInfo.VaryPeriodUnit = "D";
inDATMInfo.Reason = "Initial Treatment.";
inDATMInfo.TakehomeNum = "4";
inDATMInfo.TakehomePeriodUnit = "W";
inDATMInfo.Comment = "Ready for dosing.";
inDATMInfo.MonVisitType = "C";
inDATMInfo.TueVisitType = "C";
inDATMInfo.WedVisitType = "C";
inDATMInfo.ThuVisitType = "C";
inDATMInfo.FriVisitType = "C";
inDATMInfo.SatVisitType = "H";
inDATMInfo.SunVisitType = "H";

DATMInfoWSC outDATMInfo = new DATMInfoWSC();

outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);

if (outDATMInfo.MessageId.Length > 0)
{
    txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
}

wsDATMWebService = null;
}
}
```

5.2.2 Update Medicine Order

When a medicine order is updated, the sending OTP can send only those date elements that were changed or all of the data elements whether or not they changed. If the original medicine order has a value in a particular field, say comment, and the medicine order update has no value in comment, then the original comment will be blanked out. In other words, nothing will overwrite something.

The only fields of a medicine order that can be updated are the expiration date, next appointment date, and comments, and only if the medicine order is current. All other updates will result in the system copying the information in the medicine order to be updated to a new medicine order and the updates applied to the new medicine order. If the original medicine order to be updated was current, then it will be expired.

If there's a request to update a medicine order that doesn't exist, the medicine order will be created.

5.2.2.1 Data Transaction Fields

Table 5. Update Medicine Order – Data Transaction Fields

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|----------------------------|--|--------------------------------------|-----------|----------|---|
| INPUT AUTHORIZATION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| OR | | | | | |
| GUILD | Session Ticket | | String | Yes | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| INPUT TRANSACTION | | | | | |
| TTYP | Transaction type for medicine order | "MO" | String | Yes | |
| CMD | Command to create a medicine order. | "U" | String | Yes | |
| CLMOID | Clinical system medicine order ID | | String | Yes | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | D-ATM patient ID ten digit number | String | Yes | Example: "7124510603" |
| OTPID | The SAMHSA OTP ID creating, updating, or deleting the medicine order | Valid 8-digit SAMHSA OTP ID | String | Yes | Example: LESC's OTPID is "NY10153M" |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|------------|--|------------------|-----------|----------|---|
| ExpiraDt | Medicine order expiration date. The date after which the medicine order is no longer valid. | | String | Yes | Format: mm/dd/yyyy Example: "04/05/2007" |
| NxtapptDt | Next medical appointment date for continuing medicine order. The date of the next appointment related to the patient's medicine order for a controlled substance (e.g., Methadone), and not for other medications. | | String | | Format: mm/dd/yyyy Example: "04/05/2007" |
| Comment | Medicine Order Notes/Comment. | | String | | |

5.2.2.2 System Messages

Table 6. Update Medicine Order – System Messages

| Message ID | Message Name | Message Text |
|------------|--|---|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |
| 1002 | D-ATM Patient Not Enrolled At OTP To Create/Update/Delete Medicine Order | A medicine order transaction was attempted for a patient that wasn't enrolled in the OTP; therefore, the request was ignored. Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |

| Message ID | Message Name | Message Text |
|------------|---|---|
| 1004 | Medicine Order Transaction Failed | <p>Request to create/update/delete medicine order in D-ATM failed.</p> <p>Error(s): [List of Errors]</p> <p>Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId]</p> |
| 1005 | Update D-ATM Medicine Order Doesn't Exist | <p>An attempt was made to update an existing medicine order for a patient at an OTP; however, the medicine order couldn't be found.</p> <p>Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId]</p> |
| 1010 | Medicine Order Transaction Completed | <p>A medicine order in D-ATM was created/updated/deleted.</p> <p>Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId]</p> |

5.2.2.3 Java Sample Code

```
import org.d_atm.pilot.api_ws.*;  
  
public class UpdateMO  
{  
    public static void main(String [] args)  
    {  
        try  
        {  
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();  
  
            inDATMInfo.setLogin("OTP1");  
            inDATMInfo.setPassword("Test1234");  
  
            // Or use a given session ticket GUID  
            // inDATMInfo.setGUID("A5ED32EF-D6D3-46E2-B3AD-6B8022F20875");  
  
            inDATMInfo.setTTYP("MO");  
            inDATMInfo.setCMD("U");  
            inDATMInfo.setCLMOID("257703-15");  
            inDATMInfo.setDATMPatientId("7124510603");  
            inDATMInfo.setOTPId("NY60001M");  
            inDATMInfo.setExpiraDt("06/10/2007");  
            inDATMInfo.setNxtapptDt("06/11/2007");  
            inDATMInfo.setComment("Update Medicine Order Expiration and Next Appointment Dates.");  
  
            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();  
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();  
  
            DATMInfoWSC outDATMInfo = new DATMInfoWSC();  
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);  
  
            String message = "";  
  
            if (outDATMInfo.getMessageId().length() > 0) {  
                message = outDATMInfo.getMessageText();  
                System.out.println(message);  
            }  
            catch(Exception e)  
            {  
                System.out.println(e.getMessage());  
            }  
        }  
    }  
}
```

5.2.2.4 VB.NET Sample Code

```
Imports VBClient.DATMWS  
Imports VBClient.ClientForm  
Imports System  
Imports System.IO  
Imports System.Data  
Imports System.Xml
```

Public Class ClientForm

```

Private Sub btnUpdateMO_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles btnUpdateMO.Click
    Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()

    Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()
    inDATMInfo.Login = "OTP1"
    inDATMInfo.Password = "Test1234"

    ' Or use a given session ticket GUID
    ' inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875"

    inDATMInfo.TTYP = "MO"
    inDATMInfo.CMD = "U"
    inDATMInfo.CLMOID = "257703-15"
    inDATMInfo.DATMPatientId = "7124510603"
    inDATMInfo.OTPlId = "NY60001M"
    inDATMInfo.ExpirDt = "06/10/2007"
    inDATMInfo.NxtapptDt = "06/11/2007"
    inDATMInfo.Comment = "Update Medicine Order Expiration and Next Appointment Dates."

    Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()
    outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

    If Not (outDATMInfo.MessageId = "") Then
        txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, System.Environment.NewLine)
    End If

    wsDATMWebService = Nothing
End Sub

```

End Class

5.2.2.5 C# Sample Code

```

using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }
    }
}

```

```
}

private void btnUpdateMO_Click(object sender, EventArgs e)
{
    DATMAPIWS wsDATMWebService = new DATMAPIWS();

    DATMInfoWSC inDATMInfo = new DATMInfoWSC();

    inDATMInfo.Login = "OTP1";
    inDATMInfo.Password = "Test1234";

    // Or use a given session ticket GUID
    // inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875";

    inDATMInfo.TTYP = "MO";
    inDATMInfo.CMD = "U";
    inDATMInfo.CLMOID = "257703-15";
    inDATMInfo.DATMPatientId = "7124510603";
    inDATMInfo.OTPlId = "NY60001M";
    inDATMInfo.ExpiraDt = "06/10/2007";
    inDATMInfo.NxtapptDt = "06/11/2007";
    inDATMInfo.Comment = "Update Medicine Order Expiration and Next Appointment Dates.';

    DATMInfoWSC outDATMInfo = new DATMInfoWSC();

    outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);

    if (outDATMInfo.MessageId.Length > 0)
    {
        txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
    }

    wsDATMWebService = null;
}
}
```

5.2.3 Delete Medicine Order

This is to remove a medicine order that was created in error. A medicine order for which there are associated completed dosing events can't be removed. If a transaction is received to remove a medicine order for which there are completed dosing events, the medicine order will be expired instead.

5.2.3.1 Data Transaction Fields

Table 7. Delete Medicine Order – Data Transaction Fields

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|----------------------------|--|--------------------------------------|-----------|----------|--|
| INPUT AUTHORIZATION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| OR | | | | | |
| GUID | Session Ticket | | String | Yes | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| INPUT TRANSACTION | | | | | |
| TTYP | Transaction type for medicine order | "MO" | String | Yes | |
| CMD | Command to create a medicine order. | "D" | String | Yes | |
| CLMOID | Clinical system medicine order ID | | String | Yes | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | D-ATM patient ID ten digit number | String | Yes | Example: "7124510603" |
| OTPId | The SAMHSA OTP ID creating, updating, or deleting the medicine order | Valid 8-digit SAMHSA OTP ID | String | Yes | Example: LESC's OTPId is "NY10153M" |

5.2.3.2 System Messages

Table 8. Delete Medicine Order – System Messages

| Message ID | Message Name | Message Text |
|------------|--|--|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |
| 1002 | D-ATM Patient Not Enrolled At OTP To Create/Update/Delete Medicine Order | A medicine order transaction was attempted for a patient that wasn't enrolled in the OTP; therefore, the request was ignored. Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |
| 1004 | Medicine Order Transaction Failed | Request to create/update/delete medicine order in D-ATM failed. Error(s): [List of Errors] Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |
| 1006 | Delete D-ATM Medicine Order Doesn't Exist | An attempt was made to delete an existing medicine order for a patient at an OTP; however, the medicine order couldn't be found. Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |

| Message ID | Message Name | Message Text |
|------------|---|---|
| 1007 | Delete Current D-ATM Medicine Order With Completed Dosing | <p>An attempt was made to delete an existing medicine order for a patient at an OTP; however, the medicine order is current (today's date is between the effective and expiration dates) and has completed dosing events associated with it. Therefore, the existing medicine order was expired instead by setting the expiration date to today's date.</p> <p>Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId]</p> |
| 1008 | Delete D-ATM Medicine Order With Completed Dosing | <p>An attempt was made to delete an existing medicine order for a patient at an OTP; however, the medicine order has completed dosing events associated with it. Therefore, the medicine order cannot be deleted.</p> <p>Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId]</p> |
| 1010 | Medicine Order Transaction Completed | <p>A medicine order in D-ATM was created/updated/deleted.</p> <p>Date/Time: [Current Date and Time of Transaction] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId]</p> |

5.2.3.3 Java Sample Code

```

import org.d_atm.pilot.api_ws.*;

public class DeleteMO
{
    public static void main(String [] args)
    {
        try
        {
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.setLogin("OTP1");
            inDATMInfo.setPassword("Test1234");

            // Or use a given session ticket GUID
            // inDATMInfo.setGUID("A5ED32EF-D6D3-46E2-B3AD-6B8022F20875");

            inDATMInfo.setTTYP("MO");
            inDATMInfo.setCMD("D");
            inDATMInfo.setCLMOID("257703-15");
            inDATMInfo.setDATMPatientId("7124510603");
            inDATMInfo.setOTPId("NY60001M");

            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();

            DATMInfoWSC outDATMInfo = new DATMInfoWSC();
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);

            String message = "";

            if (outDATMInfo.getMessageId().length() > 0) {
                message = outDATMInfo.getMessageText();
                System.out.println(message);
            }
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}

```

5.2.3.4 VB.NET Sample Code

```

Imports VBClient.DATMWS
Imports VBClient.ClientForm
Imports System
Imports System.IO
Imports System.Data
Imports System.Xml

Public Class ClientForm

    Private Sub btnDeleteMO_Click(ByVal sender As System.Object, _

```

```

    ByVal e As System.EventArgs) Handles btnDeleteMO.Click
    Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()

    Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()

    inDATMInfo.Login = "OTP1"
    inDATMInfo.Password = "Test1234"

    ' Or use a given session ticket GUID
    ' inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875"

    inDATMInfo.TTYP = "MO"
    inDATMInfo.CMD = "D"
    inDATMInfo.CLMOID = "257703-15"
    inDATMInfo.DATMPatientId = "7124510603"
    inDATMInfo.OPTId = "NY60001M"

    Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()

    outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

    If Not (outDATMInfo.MessageId = "") Then
        txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, System.Environment.NewLine)
    End If

    wsDATMWebService = Nothing
End Sub

End Class

```

5.2.3.5 C# Sample Code

```

using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void btnDeleteMO_Click(object sender, EventArgs e)
        {
            DATMAPIWS wsDATMWebService = new DATMAPIWS();

```

```
DATMInfoWSC inDATMInfo = new DATMInfoWSC();

inDATMInfo.Login = "OTP1";
inDATMInfo.Password = "Test1234";

// Or use a given session ticket GUID
// inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875";

inDATMInfo.TTYP = "MO";
inDATMInfo.CMD = "D";
inDATMInfo.CLMOID = "257703-15";
inDATMInfo.DATMPatientId = "7124510603";
inDATMInfo.OTPId = "NY60001M";

DATMInfoWSC outDATMInfo = new DATMInfoWSC();

outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);

if (outDATMInfo.MessageId.Length > 0)
{
    txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
}

wsDATMWebService = null;
}
}
```

5.3 Dosing Event Transactions

Using the API, completed dosing events can be created or removed.

5.3.1 Create Completed Dosing Event

Dosing events for a patient are associated with the appropriate medicine order.

If the patient isn't enrolled in D-ATM at the OTP, the transaction will be ignored and a message will be returned.

If a completed dosing event is created for which there is no medicine order or the medicine order is expired, D-ATM will automatically create a default medicine order to cover that day. A warning message to that effect will be returned.

If a completed dosing event is received for a day that already has a completed dosing event, and if the medicine order doesn't indicate a split dose, then the duplicate dosing event will be recorded and a warning message about the duplication will be returned.

5.3.1.1 Data Transaction Fields

Table 9. Create Completed Dosing Event – Data Transaction Fields

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|----------------------------|--|--------------------------------------|-----------|----------|---|
| INPUT AUTHORIZATION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| OR | | | | | |
| GULd | Session Ticket | | String | Yes | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| INPUT TRANSACTION | | | | | |
| TTYP | Transaction type for dosing event | "DE" | String | Yes | |
| CMD | Command to create a dosing event. | "C" | String | Yes | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | D-ATM patient ID ten digit number | String | Yes | Example: "7124510603" |
| OTPId | The SAMHSA OTP ID creating, updating, or deleting the dosing event | Valid 8-digit SAMHSA OTP ID | String | Yes | Example: LESC's OTPId is "NY10153M" |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|-------------------|---|--|------------------|-----------------|---|
| Medicine | Code for Methadone, Buprenorphine, or other controlled substance | “1” = Methadone “2” = Subutex “3” = Suboxone “4” = Naloxone | String | Yes | |
| VisitType | If the dose was given at a clinic visit or is a take home. | “C” = Clinic “H” = Take Home | String | Yes | |
| DoseAmt | The amount in milligrams of the dose provided. | | String | Yes | Example: “25”, “20.50”, etc... |
| SplitDoseNo | If this dose was one of a split dose for the day, then indicate if dose number 1 or 2. | “1” = First split dose “2” = Second split dose | String | Yes | |
| StatusCde | Dosing Event Status. | “D” = Dispensed | String | Yes | |
| DoseDt | Date the dose is intended to be taken. For the clinic dose, the DispensedDt and DoseDt are the same. For a take home dose, this should be the date the taken home is intended to be taken. | | String | Yes | Format: mm/dd/yyyy Example: “04/05/2007” |
| DispensedDt | Dosing Event Dispensed Date. For clinic dose, the DispensedDt and DoseDt are equal. For take home dose, this should be the date the take home was given to the patient as opposed to the date the patient is supposed to take the dose which is DoseDt. | | String | Yes | Format: mm/dd/yyyy Example: “04/05/2007” |
| DispensedTm | The time of dosing dispensed. | | String | Yes | Format: hh:mmAM/PM Example: “10:00AM” |
| TakehomeNum | The number of take home doses provided to the patient at the visit. | | String | | |
| Comment | Dosing Event Notes/Comment. | | String | | |

5.3.1.2 System Messages

Table 10. Create Completed Dosing Event – System Messages

| Message ID | Message Name | Message Text |
|------------|--|---|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |
| 1011 | D-ATM Patient Not Enrolled At OTP To Create/Update/Delete Dosing Event | A dosing event transaction was attempted for a patient that wasn't enrolled in the OTP; therefore, the request was ignored. Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] Date/Time of Dosage Dispensed: [Date and Time of dosage dispensing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId] |
| 1012 | Dosing Event Transaction Failed | Request to create/update/delete dosing event in D-ATM failed.. Error(s): [List of Errors] Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] Date/Time of Dosage Dispensed: [Date and Time of dosage dispensing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId] |
| 1013 | Medicine Order Not Current For Dosing Event | A dosing event update was attempted for a patient that doesn't have a current medicine order at the OTP. Therefore, a medicine order was created for the patient for the dosing event date. Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] Date/Time of Dosage Dispensed: [Date and Time of dosage dispensing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId] |

| Message ID | Message Name | Message Text |
|------------|---|--|
| 1014 | Medicine Order Not Found For Dosing Event | <p>A dosing event update was attempted for a patient that doesn't have a medicine order at the OTP. Therefore, a medicine order was created for the patient for the dosing event date.</p> <p>Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] Date/Time of Dosage Dispensed: [Date and Time of dosage dispensing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId]</p> |
| 1015 | D-ATM Dosing Event Duplication | <p>A completed dosing event in D-ATM was created for a patient who already had a completed dosing event for that day.</p> <p>Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] Date/Time of Dosage Dispensed: [Date and Time of dosage dispensing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId]</p> |
| 1016 | Medicine Order Warning To Dosing OTP | <p>A dosing event update was performed for a patient with a current medicine order at other OTP(s). Any OTP(s) with a current D-ATM medicine order for this patient will be alerted to this dosing event.</p> <p>Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] Date/Time of Dosage Dispensed: [Date and Time of dosage dispensing event] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId]</p> |

| Message ID | Message Name | Message Text |
|------------|------------------------------------|---|
| 1018 | Dosing Event Transaction Completed | A completed dosing event in D-ATM was created/updated/deleted. Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] Date/Time of Dosage Dispensed: [Date and Time of dosage dispensing event] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |

5.3.1.3 Java Sample Code

```
import org.d_atm.pilot.api_ws.*;  
  
public class CreateDose  
{  
    public static void main(String [] args)  
    {  
        try  
        {  
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();  
  
            inDATMInfo.setLogin("OTP1");  
            inDATMInfo.setPassword("Test1234");  
  
            // Or use a given session ticket GUID  
            // inDATMInfo.setGUID("A5ED32EF-D6D3-46E2-B3AD-6B8022F20875");  
  
            inDATMInfo.setTTYP("DE");  
            inDATMInfo.setCMD("C");  
            inDATMInfo.setDATMPatientId("7124510603");  
            inDATMInfo.setOTPID("NY60001M");  
            inDATMInfo.setMedicine("1");  
            inDATMInfo.setVisitType("C");  
            inDATMInfo.setDoseAmt("25");  
            inDATMInfo.setSplitDoseNo("");  
            inDATMInfo.setStatusCde("D");  
            inDATMInfo.setDoseDt("04/20/2007");  
            inDATMInfo.setDispensedDt("04/20/2007");  
            inDATMInfo.setDispensedTm("11:00AM");  
            inDATMInfo.setTakehomeNum("2");  
            inDATMInfo.setComment("Dosing Patient.");  
  
            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();  
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();  
  
            DATMInfoWSC outDATMInfo = new DATMInfoWSC();  
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);  
  
            String message = "";  
  
            if (outDATMInfo.getMessageId().length() > 0) {  
                message = outDATMInfo.getMessageText();  
                System.out.println(message);  
            }  
        }  
        catch(Exception e)  
        {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

5.3.1.4 VB.NET Sample Code

```
Imports VBClient.DATMWS
Imports VBClient.ClientForm
Imports System
Imports System.IO
Imports System.Data
Imports System.Xml

Public Class ClientForm

    Private Sub btnCreateDose_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnCreateDose.Click
        Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()

        Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()

        inDATMInfo.Login = "OTP1"
        inDATMInfo.Password = "Test1234"

        ' Or use a given session ticket GUID
        ' inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875"

        inDATMInfo.TTYP = "DE"
        inDATMInfo.CMD = "C"
        inDATMInfo.DATMPatientId = "7124510603"
        inDATMInfo.OPTId = "NY60001M"
        inDATMInfo.Medicine = "1"
        inDATMInfo.VisitType = "C"
        inDATMInfo.DoseAmt = "25"
        inDATMInfo.SplitDoseNo = ""
        inDATMInfo.StatusCde = "D"
        inDATMInfo.DoseDt = "04/20/2007"
        inDATMInfo.DispensedDt = "04/20/2007"
        inDATMInfo.DispensedTm = "11:00AM"
        inDATMInfo.TakehomeNum = "2"
        inDATMInfo.Comment = "Dosing Patient."

        Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()
        outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

        If Not (outDATMInfo.MessageId = "") Then
            txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, System.Environment.NewLine)
        End If

        wsDATMWebService = Nothing
    End Sub

End Class
```

5.3.1.5 C# Sample Code

```
using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void btnCreateDose_Click(object sender, EventArgs e)
        {
            DATMAPIWS wsDATMWebService = new DATMAPIWS();

            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.Login = "OTP1";
            inDATMInfo.Password = "Test1234";

            // Or use a given session ticket GUID
            // inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875";

            inDATMInfo.TTYP = "DE";
            inDATMInfo.CMD = "C";
            inDATMInfo.DATMPatientId = "7124510603";
            inDATMInfo.OPTId = "NY60001M";
            inDATMInfo.Medicine = "1";
            inDATMInfo.VisitType = "C";
            inDATMInfo.DoseAmt = "25";
            inDATMInfo.SplitDoseNo = "";
            inDATMInfo.StatusCde = "D";
            inDATMInfo.DoseDt = "04/20/2007";
            inDATMInfo.DispensedDt = "04/20/2007";
            inDATMInfo.DispensedTm = "11:00AM";
            inDATMInfo.TakehomeNum = "2";
            inDATMInfo.Comment = "Dosing Patient.';

            DATMInfoWSC outDATMInfo = new DATMInfoWSC();

            outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);

            if (outDATMInfo.MessageId.Length > 0)
            {
                txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
            }
        }
    }
}
```

```
    }  
  
    wsDATMWebService = null;  
}  
}
```

5.3.2 Delete Completed Dosing Event

To remove a completed dosing event that has been sent in error, there should already be a completed dosing event in D-ATM for that patient for that day. If not, a warning message will be returned.

5.3.2.1 Data Transaction Fields

Table 11. Delete Completed Dosing Event – Data Transaction Fields

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|----------------------------|--|--------------------------------------|-----------|----------|---|
| INPUT AUTHORIZATION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| OR | | | | | |
| GUID | Session Ticket | | String | Yes | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| INPUT TRANSACTION | | | | | |
| TTYP | Transaction type for dosing event | "DE" | String | Yes | |
| CMD | Command to delete a dosing event. | "D" | String | Yes | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | D-ATM patient ID ten digit number | String | Yes | Example: "7124510603" |
| OTPId | The SAMHSA OTP ID creating, updating, or deleting the dosing event | Valid 8-digit SAMHSA OTP ID | String | Yes | Example: LESC's OTPId is "NY10153M" |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|------------|-----------------------------|------------------|-----------|----------|---|
| DoseDt | Date the dose to be deleted | | String | Yes | Format: mm/dd/yyyy Example: "04/05/2007" |

5.3.2.2 System Messages

Table 12. Delete Completed Dosing Event – System Messages

| Message ID | Message Name | Message Text |
|------------|--|---|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |
| 1011 | D-ATM Patient Not Enrolled At OTP To Create/Update/Delete Dosing Event | A dosing event transaction was attempted for a patient that wasn't enrolled in the OTP; therefore, the request was ignored. Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId] |
| 1012 | Dosing Event Transaction Failed | Request to create/update/delete dosing event in D-ATM failed.. Error(s): [List of Errors] Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId] |
| 1017 | Delete D-ATM Completed Dosing Event That Does Not Exist | An attempt was made to delete an existing completed dosing event at an OTP; however, the completed dosing event couldn't be found and was therefore not deleted. Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] OTP ID: [OTPId] D-ATM Patient ID: [DATMPatientId] |

| Message ID | Message Name | Message Text |
|------------|------------------------------------|--|
| 1018 | Dosing Event Transaction Completed | A completed dosing event in D-ATM was created/updated/deleted. Date/Time: [Current Date and Time of Transaction] Date of Dosage: [Date of dosing event] OTP ID: [OTPId] Clinical System Medicine Order ID: [CLMOID] D-ATM Patient ID: [DATMPatientId] |

5.3.2.3 Java Sample Code

```
import org.d_atm.pilot.api_ws.*;

public class DeleteDose
{
    public static void main(String [] args)
    {
        try
        {
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.setLogin("OTP1");
            inDATMInfo.setPassword("Test1234");

            // Or use a given session ticket GUID
            // inDATMInfo.setGULd("A5ED32EF-D6D3-46E2-B3AD-6B8022F20875");

            inDATMInfo.setTTYP("DE");
            inDATMInfo.setCMD("D");
            inDATMInfo.setDATMPatientId("7124510603");
            inDATMInfo.setOTPId("NY60001M");
            inDATMInfo.setDoseDt("04/20/2007");

            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();

            DATMInfoWSC outDATMInfo = new DATMInfoWSC();
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);

            String message = "";

            if (outDATMInfo.getMessageId().length() > 0) {
                message = outDATMInfo.getMessageText();
                System.out.println(message);
            }
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

5.3.2.4 VB.NET Sample Code

```
Imports VBClient.DATMWS
Imports VBClient.ClientForm
Imports System
Imports System.IO
Imports System.Data
Imports System.Xml
```

```
Public Class ClientForm
```

```
    Private Sub btnDeleteDose_Click(ByVal sender As System.Object, _
```

```

    ByVal e As System.EventArgs) Handles btnDeleteDose.Click
    Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()

    Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()

    inDATMInfo.Login = "OTP1"
    inDATMInfo.Password = "Test1234"

    ' Or use a given session ticket GUID
    ' inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875"

    inDATMInfo.TTYP = "DE"
    inDATMInfo.CMD = "D"
    inDATMInfo.DATMPatientId = "7124510603"
    inDATMInfo.OTPlId = "NY60001M"
    inDATMInfo.DoseDt = "04/20/2007"

    Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()

    outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

    If Not (outDATMInfo.MessageId = "") Then
        txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, System.Environment.NewLine)
    End If

    wsDATMWebService = Nothing
End Sub

End Class

```

5.3.2.5 C# Sample Code

```

using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void btnDeleteDose_Click(object sender, EventArgs e)
        {
            DATMAPIWS wsDATMWebService = new DATMAPIWS();

```

```

DATMInfoWSC inDATMInfo = new DATMInfoWSC();

inDATMInfo.Login = "OTP1";
inDATMInfo.Password = "Test1234";

// Or use a given session ticket GUID
// inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875";

inDATMInfo.TTYP = "DE";
inDATMInfo.CMD = "D";
inDATMInfo.DATMPatientId = "7124510603";
inDATMInfo.OTPid = "NY60001M";
inDATMInfo.DoseDt = "04/20/2007";

DATMInfoWSC outDATMInfo = new DATMInfoWSC();

outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);

if (outDATMInfo.MessageId.Length > 0)
{
    txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
}

wsDATMWebService = null;
}
}

```

5.4 Interface Transactions from D-ATM to Clinical System

5.4.1 Dose Administered to Patient with a Current Medicine Order at the Requesting OTP by Another OTP

OTPs have an interest in dosing events administered by other OTPs to their patients (patients for whom the OTP has a current medicine order). Therefore, whenever D-ATM detects that a dose was administered to a patient that has a current medicine order at another OTP, this event is recorded for the other OTPs in its ‘queue of patient visit events occurring at other OTPs.’

Note that while OTPs may receive an email message concerning an event for one of its patients at another OTP, the email will not include confidential details. These details are recorded in the system in the OTP’s queue of patient visit events occurring at other OTPs.

The queue of patient visit events stores summary information about dosing events. For example, if a patient receives a clinic dose and a number of take home doses from another clinic, the queue will reflect the clinic dose and the number of take home doses dispensed. The queue does not store an individual record for individual take home doses dispensed. Therefore, when D-ATM receives a request for doses administered by another clinic to a patient with a current medicine order at the requesting clinic, it returns records stored in the dosing event table rather than the queue. This way, individual clinic and take homes doses that were dispensed will be returned to the requesting clinic.

The records returned will be dosing events administered to a patient by an OTP other than the requesting OTP where the requesting OTP has or had a current medicine order for the patient at the time of the dosing event.

5.4.1.1 Data Transaction Fields

Table 13. Dose Administered to Patient w/ Current Medicine Order at Requesting OTP by Another OTP – Data Transaction Fields

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|----------------------------|--|--------------------------------------|-----------|----------|---|
| INPUT AUTHORIZATION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| OR | | | | | |
| GUID | Session Ticket | | String | Yes | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| INPUT TRANSACTION | | | | | |
| TTYP | Transaction type for dosing event | "DE" | String | Yes | |
| CMD | Command to retrieve dosing events administered by another clinic to a patient with a current medicine order at the requesting clinic. | "R" | String | Yes | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | D-ATM patient ID ten digit number | String | Yes | Example: "7124510603" |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|---------------------------|--|-----------------------------|-----------|----------|-------------------------------------|
| OTPId | The SAMHSA OTP ID requesting transaction. | Valid 8-digit SAMHSA OTP ID | String | Yes | Example: LESC's OTPId is "NY10153M" |
| OUTPUT TRANSACTION | | | | | |
| DATMList | Table records in XML representation of guest dosing information administered to a patient. The data fields are defined under OUTPUT XML LISTING. | | String | | |
| OUTPUT XML LISTING | | | | | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | | String | | |
| DispensedDt | Date the dose was dispensed to patient. | | String | | |
| Medicine | Medicine dispensed to patient. | | String | | |
| VisitType | If the dose was given at a clinic visit or is a take home. | | String | | |
| DoseAmt | The amount in milligrams of the dose provided. | | String | | |
| StatusCde | Dosing Event status code. | | String | | |
| TakehomeNum | The number of take home doses provided to the patient at the visit. | | String | | |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|---------------|--|------------------|-----------|----------|--------------|
| Comment | Dosing Event Notes/Comment. | | String | | |
| DosingOTPName | The name of the OTP that provided the dose. | | String | | |
| DosingOTPid | The 8-digit SAMHSA of the OTP that provided the dose. | | String | | |
| ContactName | The name of the staff member at the OTP to contact. | | String | | |
| ContactPhone | The phone number of the staff member at the OTP to contact. | | String | | |
| ContactEmail | The email address of the staff member at the OTP to contact. | | String | | |
| Home OTPid | The 8-digit SAMHSA ID of the patient's Home OTP. | | String | | |
| HomeOTPName | The name of the patient's Home OTP. | | String | | |

5.4.1.2 System Message

Table 14. Dose Administered to Patient w/ Current Medicine Order at the Requesting OTP by Another OTP – System Messages

| Message ID | Message Name | Message Text |
|------------|---------------|--|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |

5.4.1.3 Java Sample Code

```
import org.apache.xerces.parsers.DOMParser;
import org.w3c.dom.*;
import org.xml.sax.*;
import org.d_atm.pilot.api_ws.*;

public class GetDoseList
{
    public static void main(String [] args)
    {
        try
        {
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.setLogin("OTP1");
            inDATMInfo.setPassword("Test1234");

            // Or use a given session ticket GUID
            // inDATMInfo.setGUID("A5ED32EF-D6D3-46E2-B3AD-6B8022F20875");

            inDATMInfo.setTTYP("DE");
            inDATMInfo.setCMD("R");
            inDATMInfo.setDATMPatientId("7124510603");
            inDATMInfo.setOTPId("NY60001M");

            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();

            DATMInfoWSC outDATMInfo = new DATMInfoWSC();
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);

            String message = "";

            if (outDATMInfo.getMessageId().length() > 0) {
                message = outDATMInfo.getMessageText();
                System.out.println(message);
            }
            else {
                String xmlList = outDATMInfo.getDATMList();
                System.out.println(xmlList);

                DOMParser parser = new DOMParser();
                parser.parse(new InputSource(new java.io.StringReader(xmlList)));
                Document doc = parser.getDocument();
                doc.getDocumentElement().normalize ();
                System.out.println ("Root element is " + doc.getDocumentElement().getNodeName() + ".");

                NodeList listOfRecords = doc.getElementsByTagName("DoseEvent");
                int totalPersons = listOfRecords.getLength();
                System.out.println("Total number of Table Records: " + totalPersons);

                for (int s=0; s<listOfRecords.getLength(); s++) {
                    Node firstNode = listOfRecords.item(s);
                    if (firstNode.getNodeType() == Node.ELEMENT_NODE) {
```

```
Element firstElement = (Element)firstNode;
System.out.println("*****");
try {
    NodeList nodeList1 = firstElement.getElementsByTagName("DATMPatientId");
    Element eleNode1 = (Element)nodeList1.item(0);
    NodeList textList1 = eleNode1.getChildNodes();
    System.out.println("DATMPatientId: " +
        ((Node)textList1.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList2 = firstElement.getElementsByTagName("DispensedDt");
    Element eleNode2 = (Element)nodeList2.item(0);
    NodeList textList2 = eleNode2.getChildNodes();
    System.out.println("DispensedDt: " +
        ((Node)textList2.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList3 = firstElement.getElementsByTagName("Medicine");
    Element eleNode3 = (Element)nodeList3.item(0);
    NodeList textList3 = eleNode3.getChildNodes();
    System.out.println("Medicine: " +
        ((Node)textList3.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList4 = firstElement.getElementsByTagName("VisitType");
    Element eleNode4 = (Element)nodeList4.item(0);
    NodeList textList4 = eleNode4.getChildNodes();
    System.out.println("VisitType: " +
        ((Node)textList4.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList5 = firstElement.getElementsByTagName("DoseAmt");
```

```
Element eleNode5 = (Element)nodeList5.item(0);
NodeList textList5 = eleNode5.getChildNodes();
System.out.println("DoseAmt: " +
((Node)textList5.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
NodeList nodeList6 = firstElement.getElementsByTagName("StatusCde");
Element eleNode6 = (Element)nodeList6.item(0);
NodeList textList6 = eleNode6.getChildNodes();
System.out.println("StatusCde: " +
((Node)textList6.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
NodeList nodeList7 = firstElement.getElementsByTagName("TakehomeNum");
Element eleNode7 = (Element)nodeList7.item(0);
NodeList textList7 = eleNode7.getChildNodes();
System.out.println("TakehomeNum: " +
((Node)textList7.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
NodeList nodeList8 = firstElement.getElementsByTagName("Comment");
Element eleNode8 = (Element)nodeList8.item(0);
NodeList textList8 = eleNode8.getChildNodes();
System.out.println("Comment: " +
((Node)textList8.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
NodeList nodeList9 =
firstElement.getElementsByTagName("DosingOTPName");
Element eleNode9 = (Element)nodeList9.item(0);
NodeList textList9 = eleNode9.getChildNodes();
System.out.println("DosingOTPName: " +
((Node)textList9.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
```

```
}

try {
    NodeList nodeList10 = firstElement.getElementsByTagName("DosingOTPid");
    Element eleNode10 = (Element)nodeList10.item(0);
    NodeList textList10 = eleNode10.getChildNodes();
    System.out.println("DosingOTPid: " +
        ((Node)textList10.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList11 = firstElement.getElementsByTagName("ContactName");
    Element eleNode11 = (Element)nodeList11.item(0);
    NodeList textList11 = eleNode11.getChildNodes();
    System.out.println("ContactName: " +
        ((Node)textList11.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList12 = firstElement.getElementsByTagName("ContactPhone");
    Element eleNode12 = (Element)nodeList12.item(0);
    NodeList textList12 = eleNode12.getChildNodes();
    System.out.println("ContactPhone: " +
        ((Node)textList12.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList13 = firstElement.getElementsByTagName("ContactEmail");
    Element eleNode13 = (Element)nodeList13.item(0);
    NodeList textList13 = eleNode13.getChildNodes();
    System.out.println("ContactEmail: " +
        ((Node)textList13.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList14 =
firstElement.getElementsByTagName("HomeOTPName");
    Element eleNode14 = (Element)nodeList14.item(0);
    NodeList textList14 = eleNode14.getChildNodes();
```

```
        System.out.println("HomeOTPName: " +
            ((Node)textList14.item(0)).getNodeValue().trim());
    }
    catch(Exception e)
    {
        }

    try {
        NodeList nodeList15 = firstElement.getElementsByTagName("HomeOTPId");
        Element eleNode15 = (Element)nodeList15.item(0);
        NodeList textList15 = eleNode15.getChildNodes();
        System.out.println("HomeOTPId: " +
            ((Node)textList15.item(0)).getNodeValue().trim());
    }
    catch(Exception e)
    {
        }

    }
}

}
catch(Exception e)
{
    System.out.println(e.getMessage());
}
}
```

5.4.1.4 VB.NET Sample Code

```
Imports VBClient.DATMWS  
Imports VBClient.ClientForm  
Imports System  
Imports System.IO  
Imports System.Data  
Imports System.Xml
```

Public Class ClientForm

```
Private Sub btnGetDoseList_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles btnGetDoseList.Click  
    Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()
```

Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()

```
inDATMInfo.Login = "OTP1"  
inDATMInfo.Password = "Test1234"
```

' Or use a given session ticket GUID
'inDATMInfo.GUID = "A5ED32EE-D6D3-46E2-B3AD-6B8022E20875"

```
inDATMInfo.TTYP = "DE"  
inDATMInfo.CMD = "R"  
inDATMInfo.DATMPatientId = "7124510603"
```

```
inDATMInfo.OTPId = "NY60001M"

Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()

outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

If Not (outDATMInfo.MessageId = "") Then
    txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, _
                                                System.Environment.NewLine)
Else
    If Not (outDATMInfo.DATMList = "") Then
        Dim inStringReader As StringReader = Nothing
        Dim inDataSet As DataSet = New DataSet()
        Dim inDataTable As DataTable
        Dim sMessageOut As String = ""

        Try
            inStringReader = New StringReader(outDATMInfo.DATMList)
            inDataSet.ReadXml(inStringReader)

            For Each inDataTable In inDataSet.Tables
                sMessageOut = sMessageOut & "Tabe Name: " & _
                             inDataTable.TableName.ToString() & System.Environment.NewLine
                sMessageOut = sMessageOut & System.Environment.NewLine

                Dim inDataCol As DataColumn
                Dim inDataRow As DataRow

                sMessageOut = sMessageOut & "|"

                For Each inDataCol In inDataTable.Columns
                    sMessageOut = sMessageOut & inDataCol.ColumnName.ToString() & " | "
                Next
                sMessageOut = sMessageOut & System.Environment.NewLine

                For Each inDataRow In inDataTable.Rows
                    Dim inVal As Object

                    sMessageOut = sMessageOut & System.Environment.NewLine
                    sMessageOut = sMessageOut & "|"

                    For Each inVal In inDataRow.ItemArray
                        sMessageOut = sMessageOut & inVal.ToString() & " | "
                    Next

                    sMessageOut = sMessageOut & System.Environment.NewLine
                Next

                sMessageOut = sMessageOut & System.Environment.NewLine
                sMessageOut = sMessageOut & "Number of Rows: " & _
                             inDataTable.Rows.Count.ToString() & System.Environment.NewLine
            Next inDataTable

            Catch ex As Exception
                sMessageOut = "Error: " & ex.ToString()
            Finally
        End Try
    End If
End If
```

```

If Not inStringReader Is Nothing Then
    inStringReader.Close()
End If
End Try

txtMessage.Text = sMessageOut
End If
End If

wsDATMWebService = Nothing
End Sub

End Class

```

5.4.1.5 C# Sample Code

```

using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void btnGetDoseList_Click(object sender, EventArgs e)
        {
            DATMAPIWS wsDATMWebService = new DATMAPIWS();

            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.Login = "OTP1";
            inDATMInfo.Password = "Test1234";

            // Or use a given session ticket GUID
            // inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875";

            inDATMInfo.TTYP = "DE";
            inDATMInfo.CMD = "R";
            inDATMInfo.DATMPatientId = "7124510603";
            inDATMInfo.OTPid = "NY60001M";

            DATMInfoWSC outDATMInfo = new DATMInfoWSC();

            outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);
        }
    }
}

```

```
if (outDATMInfo.MessageId.Length > 0)
{
    txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
}
else
{
    if (outDATMInfo.DATMList.Length > 0)
    {
        StringReader inStringReader = null;
        DataSet inDataSet = new DataSet();
        String sMessageOut = "";

        try
        {
            inStringReader = new StringReader(outDATMInfo.DATMList);
            inDataSet.ReadXml(inStringReader);

            foreach(DataTable inDataTable in inDataSet.Tables)
            {
                sMessageOut = sMessageOut + "Tabe Name: " +
                    inDataTable.TableName.ToString() + System.Environment.NewLine;
                sMessageOut = sMessageOut + System.Environment.NewLine;

                sMessageOut = sMessageOut + "| ";

                foreach(DataColumn inDataCol in inDataTable.Columns)
                {
                    sMessageOut = sMessageOut + inDataCol.ColumnName.ToString() + " | ";
                }

                sMessageOut = sMessageOut + System.Environment.NewLine;

                foreach (DataRow inDataRow in inDataTable.Rows)
                {
                    sMessageOut = sMessageOut + System.Environment.NewLine;
                    sMessageOut = sMessageOut + "| ";

                    foreach (Object inVal in inDataRow.ItemArray)
                    {
                        sMessageOut = sMessageOut + inVal.ToString() + " | ";
                    }

                    sMessageOut = sMessageOut + System.Environment.NewLine;
                }

                sMessageOut = sMessageOut + System.Environment.NewLine;
                sMessageOut = sMessageOut + "Number of Rows: " +
                    inDataTable.Rows.Count.ToString() + System.Environment.NewLine;
            }
        }
        catch (Exception ex)
        {
            sMessageOut = "Error: " + ex.ToString();
        }
    finally
```

```

    {
        if (inStringReader != null)
        {
            inStringReader.Close();
        }
    }

    txtMessage.Text = sMessageOut;
}
}

wsDATMWebService = null;
}
}

```

5.4.2 Guest Retrieval Events and Doses Administered to a Patient with a Current Medicine Order by Another OTP

OTPs also have an interest in knowing when another OTP retrieves dosing information for one of its patients. For this reason, whenever D-ATM detects that an OTP retrieved dosing information for a guest patient, this event is recorded for the home OTP in its queue of patient visit events occurring at other OTPs. However, once the patient is enrolled in D-ATM as a guest patient at the guest OTP, the retrieval events are no longer recorded for the home OTP. Dosing events administered by guest OTPs are still recorded.

When an OTP other than the patient's home OTP retrieves dosing information for a patient, the home OTP will receive an informational message. The purpose is to let the home OTP know when another OTP retrieves or attempts to retrieve dosing information for one of its patients. Once the patient is enrolled as a guest in the other OTP, these messages are no longer sent to the home OTP and the retrieval events are no longer recorded in the home OTPs queue of patient visit events. (Dosing events, however, are still recorded.)

Whenever a dosing event is recorded by D-ATM, the system checks if this patient has a current medicine order at another OTP (home or guest clinic). If so, this event is recorded in the queue of these other clinics with a current medicine order for the patient. Also, the other clinics are sent informational messages. The purpose is to let the clinics with a current medicine order for a patient know when this patient is dosed by another clinic.

Note that the queue dosing events are a summary of the doses received. For example, if the patient receives a clinic dose and a number of take home doses, the queue stores the clinic dose and the number of take homes dispensed. The queue does not store the details of each take home dose dispensed. To retrieve a complete list of dosing events, including each clinic and take home dose, request doses administered to a patient with a current medicine order at the requesting OTP described in the previous section. These dosing events will be returned from the dosing event table as opposed to the queue of patient visit events.

The following transactions represent data sent from D-ATM to the clinical system when the clinical system issues a request to D-ATM to retrieve patient visit events occurring at other OTPS.

It is recommended that the clinical system be enhanced to process these transactions, including allowing clinical staff to:

- View and print the patient visit events that occurred at other OTPs in a user-friendly format;
- Manually update local records to reflect dosing events that occurred at other OTPs;
- Electronically update local records to reflect dosing events that occurred at other OTPs.

The transactions that are intended to be communicated from D-ATM to the clinical system include the two types of patient visit events that occur at other OTPs:

- Dose administered to a patient with a current medicine order at the requesting OTP by another OTP;
- Retrieval by another OTP of dosing information for a home patient.

5.4.2.1 Data Transaction Fields

**Table 15. Guest Retrieval Events and Doses Administered to a Patient w/ Current Medicine Order by Another OTP
– Data Transaction Fields**

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|----------------------------|--|-----------------------------|-----------|----------|---|
| INPUT AUTHORIZATION | | | | | |
| Login | Login ID for Clinical System within D-ATM | | String | Yes | |
| Password | Password for Clinical System within D-ATM | | String | Yes | |
| OR | | | | | |
| GUILd | Session Ticket | | String | Yes | Example: "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875" |
| INPUT TRANSACTION | | | | | |
| TTYP | Transaction type for events if the requesting clinic's queue of patient visit events occurring at other OTPs | "QUE" | String | Yes | |
| CMD | Command to retrieve queue information on guest clinic retrieval events and doses administered by another clinic to a patient with a current medicine order at the requesting clinic. | "R" | String | Yes | |
| OTPId | The SAMHSA OTP ID requesting transaction. | Valid 8-digit SAMHSA OTP ID | String | Yes | Example: "NY10153M" |
| OUTPUT TRANSACTION | | | | | |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|---------------------------|--|------------------|-----------|----------|--------------|
| DATMList | Table records in XML representation of event information by another OTP on a patient. The data fields are defined under OUTPUT XML LISTING. | | String | | |
| OUTPUT XML LISTING | | | | | |
| EventDt | Date of Event | | String | | |
| DATMPatientId | D-ATM patient ID. A unique ID assigned to each patient when that patient is first enrolled in D-ATM. This ID is associated with the patient's biometric identification, e.g., data points (minutiae) from finger scan. This ID must be stored in the clinical system since this is the link between the clinical system and D-ATM. | | String | | |
| EventName | Event Name | | String | | |
| Medicine | Medicine dispensed to patient. | | String | | |
| VisitType | If the dose was given at a clinic visit or is a take home. | | String | | |
| DoseAmt | The amount in milligrams of the dose provided. | | String | | |
| DoseStatus | Dosing Event status description. | | String | | |
| DoseTakehomeNum | The number of take home doses provided to the patient at the visit. | | String | | |
| Comment | Guest Event Notes/Comment. | | String | | |
| OTPName | The name of the OTP that created the event. | | String | | |

| Field Name | Field Description | Permitted Values | Data Type | Required | Format Notes |
|--------------|--|------------------|-----------|----------|--------------|
| OTPId | The 8-digit SAMHSA of the OTP that created the event. | | String | | |
| ContactName | The name of the staff member at the OTP to contact. | | String | | |
| ContactPhone | The phone number of the staff member at the OTP to contact. | | String | | |
| ContactEmail | The email address of the staff member at the OTP to contact. | | String | | |

5.4.2.2 System Message

Table 16. Guest Retrieval Events and Doses Administered to a Patient w/ Current Medicine Order by Another OTP – System Messages

| Message ID | Message Name | Message Text |
|------------|---------------|--|
| 1001 | Access Denied | An attempt to authenticate D-ATM web service has failed. |

5.4.2.3 Java Sample Code

```
import org.apache.xerces.parsers.DOMParser;
import org.w3c.dom.*;
import org.xml.sax.*;
import org.d_atm.pilot.api_ws.*;

public class GetEventList
{
    public static void main(String [] args)
    {
        try
        {
            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.setLogin("OTP1");
            inDATMInfo.setPassword("Test1234");

            // Or use a given session ticket GUID
            // inDATMInfo.setGUID("A5ED32EF-D6D3-46E2-B3AD-6B8022F20875");

            inDATMInfo.setTTYP("QUE");
            inDATMInfo.setCMD("R");
            inDATMInfo.setOTPId("NY60001M");

            DATMAPIWSLocator locDATMAPI = new DATMAPIWSLocator();
            DATMAPIWSSoap portDATMAPI = locDATMAPI.getDATMAPIWSSoap();

            DATMInfoWSC outDATMInfo = new DATMInfoWSC();
            outDATMInfo = portDATMAPI.DATMService(inDATMInfo);

            String message = "";

            if (outDATMInfo.getMessageId().length() > 0) {
                message = outDATMInfo.getMessageText();
                System.out.println(message);
            }
            else {
                String xmlList = outDATMInfo.getDATMList();
                System.out.println(xmlList);

                DOMParser parser = new DOMParser();
                parser.parse(new InputSource(new java.io.StringReader(xmlList)));
                Document doc = parser.getDocument();
                doc.getDocumentElement().normalize ();
                System.out.println ("Root element is " + doc.getDocumentElement().getNodeName() + ".");

                NodeList listOfRecords = doc.getElementsByTagName("PatientVisitEvent");
                int totalPersons = listOfRecords.getLength();
                System.out.println("Total number of Table Records: " + totalPersons);

                for (int s=0; s<listOfRecords.getLength(); s++) {
                    Node firstNode = listOfRecords.item(s);
                    if (firstNode.getNodeType() == Node.ELEMENT_NODE) {
                        Element firstElement = (Element)firstNode;
```

```
System.out.println("*****");
try {
    NodeList nodeList1 = firstElement.getElementsByTagName("EventDt");
    Element eleNode1 = (Element)nodeList1.item(0);
    NodeList textList1 = eleNode1.getChildNodes();
    System.out.println("EventDt: " +
        ((Node)textList1.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList2 = firstElement.getElementsByTagName("DATMPatientId");
    Element eleNode2 = (Element)nodeList2.item(0);
    NodeList textList2 = eleNode2.getChildNodes();
    System.out.println("DATMPatientId: " +
        ((Node)textList2.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList3 = firstElement.getElementsByTagName("EventName");
    Element eleNode3 = (Element)nodeList3.item(0);
    NodeList textList3 = eleNode3.getChildNodes();
    System.out.println("EventName: " +
        ((Node)textList3.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList4 = firstElement.getElementsByTagName("Medicine");
    Element eleNode4 = (Element)nodeList4.item(0);
    NodeList textList4 = eleNode4.getChildNodes();
    System.out.println("Medicine: " +
        ((Node)textList4.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList5 = firstElement.getElementsByTagName("VisitType");
    Element eleNode5 = (Element)nodeList5.item(0);
```

```
        NodeList textList5 = eleNode5.getChildNodes();
        System.out.println("VisitType: " +
            ((Node)textList5.item(0)).getNodeValue().trim());
    }
    catch(Exception e)
    {
    }

    try {
        NodeList nodeList6 = firstElement.getElementsByTagName("DoseAmt");
        Element eleNode6 = (Element)nodeList6.item(0);
        NodeList textList6 = eleNode6.getChildNodes();
        System.out.println("DoseAmt: " +
            ((Node)textList6.item(0)).getNodeValue().trim());
    }
    catch(Exception e)
    {
    }

    try {
        NodeList nodeList7 = firstElement.getElementsByTagName("DoseStatus");
        Element eleNode7 = (Element)nodeList7.item(0);
        NodeList textList7 = eleNode7.getChildNodes();
        System.out.println("DoseStatus: " +
            ((Node)textList7.item(0)).getNodeValue().trim());
    }
    catch(Exception e)
    {
    }

    try {
        NodeList nodeList8 =
            firstElement.getElementsByTagName("DoseTakehomeNum");
        Element eleNode8 = (Element)nodeList8.item(0);
        NodeList textList8 = eleNode8.getChildNodes();
        System.out.println("DoseTakehomeNum: " +
            ((Node)textList8.item(0)).getNodeValue().trim());
    }
    catch(Exception e)
    {
    }

    try {
        NodeList nodeList9 = firstElement.getElementsByTagName("Comment");
        Element eleNode9 = (Element)nodeList9.item(0);
        NodeList textList9 = eleNode9.getChildNodes();
        System.out.println("Comment: " +
            ((Node)textList9.item(0)).getNodeValue().trim());
    }
    catch(Exception e)
    {
```

```
}

try {
    NodeList nodeList10 = firstElement.getElementsByTagName("OTPName");
    Element eleNode10 = (Element)nodeList10.item(0);
    NodeList textList10 = eleNode10.getChildNodes();
    System.out.println("OTPName: " +
        ((Node)textList10.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList11 = firstElement.getElementsByTagName("OTPId");
    Element eleNode11 = (Element)nodeList11.item(0);
    NodeList textList11 = eleNode11.getChildNodes();
    System.out.println("OTPId: " +
        ((Node)textList11.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList12 = firstElement.getElementsByTagName("ContactName");
    Element eleNode12 = (Element)nodeList12.item(0);
    NodeList textList12 = eleNode12.getChildNodes();
    System.out.println("ContactName: " +
        ((Node)textList12.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList13 = firstElement.getElementsByTagName("ContactPhone");
    Element eleNode13 = (Element)nodeList13.item(0);
    NodeList textList13 = eleNode13.getChildNodes();
    System.out.println("ContactPhone: " +
        ((Node)textList13.item(0)).getNodeValue().trim());
}
catch(Exception e)
{
}

try {
    NodeList nodeList14 = firstElement.getElementsByTagName("ContactEmail");
    Element eleNode14 = (Element)nodeList14.item(0);
    NodeList textList14 = eleNode14.getChildNodes();
    System.out.println("ContactEmail: " +
        ((Node)textList14.item(0)).getNodeValue().trim());
}
```

```
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

5.4.2.4 VB.NET Sample Code

```
Imports VBClient.DATMWS
Imports VBClient.ClientForm
Imports System
Imports System.IO
Imports System.Data
Imports System.Xml

Public Class ClientForm

    Private Sub btnGetEventList_Click(ByVal sender As System.Object, _
                                         ByVal e As System.EventArgs) Handles btnGetEventList.Click
        Dim wsDATMWebService As DATMAPIWS = New DATMAPIWS()

        Dim inDATMInfo As DATMInfoWSC = New DATMInfoWSC()
        inDATMInfo.Login = "OTP1"
        inDATMInfo.Password = "Test1234"

        ' Or use a given session ticket GUID
        ' inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875"

        inDATMInfo.TTYP = "QUE"
        inDATMInfo.CMD = "R"
        inDATMInfo.OPTId = "NY60001M"

        Dim outDATMInfo As DATMInfoWSC = New DATMInfoWSC()
        outDATMInfo = wsDATMWebService.DATMService(inDATMInfo)

        If Not (outDATMInfo.MessageId = "") Then
            txtMessage.Text = outDATMInfo.MessageText.Replace(ControlChars.Lf, System.Environment.NewLine)
        Else
            If Not (outDATMInfo.DATMList = "") Then
                Dim inStringReader As StringReader = Nothing
                Dim inDataSet As DataSet = New DataSet()
                Dim inDataTable As DataTable
                Dim sMessageOut As String = ""
```

```
Try
    inStringReader = New StringReader(outDATMInfo.DATMList)
    inDataSet.ReadXml(inStringReader)

    For Each inDataTable In inDataSet.Tables
        sMessageOut = sMessageOut & "Tabe Name: " & _
            inDataTable.TableName.ToString() & System.Environment.NewLine
        sMessageOut = sMessageOut & System.Environment.NewLine

        Dim inDataCol As DataColumn
        Dim inDataRow As DataRow

        sMessageOut = sMessageOut & "|"

        For Each inDataCol In inDataTable.Columns
            sMessageOut = sMessageOut & inDataCol.ColumnName.ToString() & " | "
        Next
        sMessageOut = sMessageOut & System.Environment.NewLine

        For Each inDataRow In inDataTable.Rows
            Dim inVal As Object

            sMessageOut = sMessageOut & System.Environment.NewLine
            sMessageOut = sMessageOut & "|"

            For Each inVal In inDataRow.ItemArray
                sMessageOut = sMessageOut & inVal.ToString() & " | "
            Next

            sMessageOut = sMessageOut & System.Environment.NewLine
        Next

        sMessageOut = sMessageOut & System.Environment.NewLine
        sMessageOut = sMessageOut & "Number of Rows: " & _
            inDataTable.Rows.Count.ToString() & System.Environment.NewLine
    Next inDataTable

    Catch ex As Exception
        sMessageOut = "Error: " & ex.ToString()
    Finally
        If Not inStringReader Is Nothing Then
            inStringReader.Close()
        End If
    End Try

    txtMessage.Text = sMessageOut
End If
End If

wsDATMWebService = Nothing
End Sub

End Class
```

5.4.2.5 C# Sample Code

```
using CSharpClient.DATMWS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace CSharpClient
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void btnGetEventList_Click(object sender, EventArgs e)
        {
            DATMAPIWS wsDATMWebService = new DATMAPIWS();

            DATMInfoWSC inDATMInfo = new DATMInfoWSC();

            inDATMInfo.Login = "OTP1";
            inDATMInfo.Password = "Test1234";

            // Or use a given session ticket GUID
            // inDATMInfo.GUID = "A5ED32EF-D6D3-46E2-B3AD-6B8022F20875";

            inDATMInfo.TTYP = "QUE";
            inDATMInfo.CMD = "R";
            inDATMInfo.OTPlId = "NY60001M";

            DATMInfoWSC outDATMInfo = new DATMInfoWSC();

            outDATMInfo = wsDATMWebService.DATMService(inDATMInfo);

            if (outDATMInfo.MessageId.Length > 0)
            {
                txtMessage.Text = outDATMInfo.MessageText.Replace("\n", System.Environment.NewLine);
            }
            else
            {
                if (outDATMInfo.DATMList.Length > 0)
                {
                    StringReader inStringReader = null;
                    DataSet inDataSet = new DataSet();
                    String sMessageOut = "";

                    try
                    {
```

```
inStringReader = new StringReader(outDATMInfo.DATMLIST);
inDataSet.ReadXml(inStringReader);

foreach (DataTable inDataTable in inDataSet.Tables)
{
    sMessageOut = sMessageOut + "Tabe Name: " +
        inDataTable.TableName.ToString() + System.Environment.NewLine;
    sMessageOut = sMessageOut + System.Environment.NewLine;

    sMessageOut = sMessageOut + "| ";

    foreach (DataColumn inDataCol in inDataTable.Columns)
    {
        sMessageOut = sMessageOut + inDataCol.ColumnName.ToString() + " | ";
    }

    sMessageOut = sMessageOut + System.Environment.NewLine;

    foreach (DataRow inDataRow in inDataTable.Rows)
    {
        sMessageOut = sMessageOut + System.Environment.NewLine;
        sMessageOut = sMessageOut + "| ";

        foreach (Object inVal in inDataRow.ItemArray)
        {
            sMessageOut = sMessageOut + inVal.ToString() + " | ";
        }

        sMessageOut = sMessageOut + System.Environment.NewLine;
    }

    sMessageOut = sMessageOut + System.Environment.NewLine;
    sMessageOut = sMessageOut + "Number of Rows: " +
        inDataTable.Rows.Count.ToString() + System.Environment.NewLine;
}

catch (Exception ex)
{
    sMessageOut = "Error: " + ex.ToString();
}
finally
{
    if (inStringReader != null)
    {
        inStringReader.Close();
    }
}

txtMessage.Text = sMessageOut;
}

wsDATMWebService = null;
}
```